JOÃO ROCHA DA SILVA

USAGE-DRIVEN APPLICATION PROFILE GENERATION USING ONTOLOGIES

SUPERVISOR: CRISTINA RIBEIRO (PH.D.)
CO-SUPERVISOR: JOÃO CORREIA LOPES (PH.D.)

PH.D. THESIS

Faculdade de Engenharia da Universidade do Porto /
INESC TEC
http://www.fe.up.pt
2016, May

João Rocha da Silva: *Usage-driven application profile generation using ontologies*, Ph.D. Thesis, 2016, May.

WEBSITE:
https://sigarra.up.pt/feup/pt/fest_geral.cursos_list?pv_num_unico=200405299

E-MAIL:
joaorosilva@gmail.com

ABSTRACT

Nowadays, research is driven by ever increasing amounts of datasets and supported by more and more accessible computation power. This data is expensive to produce and very diverse due to its domain-specific nature. Such domain specificity calls for the participation of data creators in the description of their datasets, since they are the most knowledgeable regarding the meaning of their data.

Capturing each dataset's production context is thus necessary to enable their reuse, either by the creators or by other researchers within their research group, as well as from external groups. Researchers are not data management experts, though, so they need assistance in the production of their metadata records. The most common approach is to adopt a fixed set of descriptors for all research domains, but those descriptors are often not enough.

This research focuses on supporting the description of research datasets by the researchers themselves, using appropriate metadata. We started by gathering requirements in collaboration with a panel of researchers. The requirements, combined with our vision of an integrated data management workflow have led to the development of several solutions which have evolved towards a platform where data can be organised and described.

The first approach towards a research data management environment was a prototype repository designed for the deposit of datasets and based on DSpace (UPData). Shifting to an earlier moment in the data production workflow, we followed it up with two complementary solutions (UPBox and DataNotes) designed to support researchers in their regular data management tasks. These solutions introduced the two aspects identified as the most important during our requirements gathering: ease of use by non-experts and high-quality, appropriate metadata.

From the lessons learned from these developments we could outline, design and build the Dendro platform, a user-friendly research data management platform that allows researchers to deposit and describe their datasets. Behind the scenes is a graph-based data model supported by Linked Open Data. It is fully built on ontologies that can be directly drawn from the web or designed according to the metadata requirements of each research domain. We have also have developed a modeling process for these lightweight ontologies, having instantiated it several times during this work.

As more of these ontologies are introduced in Dendro, researchers can be easily overwhelmed by the increasing number of descriptors available for dataset description. To cope with this information overload and assist them in building their own application profiles, we implemented a recommendation module in Dendro. This module recommends descriptors suitable for different research domains and is driven by the usage patterns of users, taking different interactions into account to select the most appropriate descriptors. Our approach was tested with a panel of researchers from 11 different domains. Results show an increase in the usage of domains-specific descriptors, as well as an easier adaptation to the process of data description by these non-expert users, while maintaining the quality of the metadata records.

RESUMO

O trabalho de investigação é actualmente baseado em conjuntos de dados cada vez mais numerosos e é suportado por capacidades computacionais cada vez maiores e mais acessíveis. A produção de tais dados é dispensiosa e a sua natureza é muito diversa, dependendo do domínio de investigação de onde esses dados provêm. Esta especificidade torna necessária a participação dos criadores na descrição dos seus conjuntos de dados, já que estes possuem um conhecimento profundo do significado dos dados que produzem.

A captura do contexto de produção de cada conjunto de dados é necessária para tornar possível a sua reutilização, tanto pelos criadores como por outros investigadores dentro e fora do grupo de investigação. Contudo, os investigadores não são peritos em gestão de dados, portanto necessitam de suporte à produção dos seus registos de metadados. A abordagem mais comum passa pela adopção de um conjunto fixo de descritores aplicado a todos os domínios de investigação, mas esses descritores são frequentemente insuficientes.

Este trabalho de investigação foca-se no suporte à descrição de conjuntos de dados de investigação por parte dos próprios investigadores, utilizando metadados adequados. Começou com uma recolha de requisitos em estreita colaboração com um painel de investigadores; estes requisitos, combinados com a nossa visão para um *workflow* integrado de gestão de dados, levaram ao desenvolvimento de diversas soluções que evoluíram no sentido de uma plataforma onde os dados podem ser organizados e descritos.

A primeira abordagem no sentido da criação de um ambiente de gestão de dados científicos consistiu num protótipo de repositório desenhado para depósito de dados e baseado na ferramenta DSpace (UPData). No sentido de introduzir a descrição de dados mais a montante no processo de criação de dados, duas soluções integradas foram também implementadas (UPBox e DataNotes). Estas soluções introduziram dois aspectos considerados muito importantes durante o processo de recolha de requisitos: a facilidade de utilização por parte de utilizadores sem conhecimento especializado e a produção de metadados apropriados e de qualidade.

As lições retiradas destes desenvolvimentos suportaram o desenho e construção da plataforma Dendro, um ambiente amigável para gestão de dados de investigação que permite aos investigadores depositar e descrever os seus conjuntos de dados. Em segundo plano existe um modelo de dados em grafo, assente em *Linked Open Data*. É completamente construído sobre ontologias que podem ser retiradas da web, ou desenhadas de acordo com os requisitos de metadados de cada grupo de investigação. Um processo de modelação destas ontologias *lightweight* foi também desenvolvido e instanciado por diversas vezes durante este trabalho.

Á medida que mais ontologias deste tipo são introduzidas na plataforma Dendro, mais difícil se torna para os investigadores escolher os descritores mais adequados para a descrição dos seus conjuntos de dados. Por forma a lidar com esta sobrecarga de informação e ajudar os investigadores a construir os seus próprios perfis de aplicação, foi implementado um módulo de recomendação na plataforma Dendro. Este módulo recomenda descritores adequados para a descrição de conjunto de dados de diferentes

domínios, e baseia-se nos padrões de utilização dos diferentes utilizadores, tendo em conta diversos tipos de interações na seleção dos descritores mais apropriados. A abordagem foi testada com um painel de investigadores provenientes de 11 domínios de investigação distintos. Os resultados obtidos mostram um aumento da utilização de descritores específicos do domínio, assim como uma adaptação mais fácil ao processo de descrição por parte destes utilizadores sem conhecimentos profundos de descrição e metadados, ao mesmo tempo que mantém a qualidade dos registos de metadados produzidos.

ACKNOWLEDGEMENTS

This work was only possible due to the support, dedication and encouragement of my supervisors, Cristina Ribeiro and João Correia Lopes, who always supported this work with invaluable and timely reviews of all publications, informed opinions and adequate guidance whenever I needed. The first and special thanks go to them.

I would also like to thank everyone who worked with me during this time. InfoLab is an awesome group both inside the laboratory and outside it.

- **João Aguiar Castro** (InfoLab[1]), whose tireless work, willingness to learn and help with every single task was superb. His work with the researchers from the early days, even before Dendro existed, was crucial for making sure that the solution was supported on the actual needs of the users. His Information Science expertise greatly complements the engineering involved in this work, with his ontology development process being instantiated several times to formalize the needs of the different research groups directly into Dendro's data model.
- **Ricardo Amorim** (InfoLab), which has the can-do attitude of a top engineer. From requirements gathering, meetings with researchers and testing with Dendro and its API, Ricardo was always willing to help. As a researcher, his skills and strong analytical ability are undeniable. From a technical standpoint, his help in the development of the recommender module, his self-taught skills with Android, enthusiasm for open source and willingness to learn make him an awesome asset to the team throughout the entire time.
- **Yulia Karimova** (InfoLab), who participated in the design of the lightweight ontologies used in Dendro, in the analysis of questionnaires, and in the meetings with researchers during the evaluation experiments.
- **Paula Fortuna** (InfoLab), who contributed with her Psychology skills and effort to the design of the user satisfaction questionnaires; her contributions in data analysis and programming in the recommendation module have also to be mentioned.
- **Filipe Perdigão** (InfoLab), who worked in the development of repository plugins which allow Dendro to deposit datasets to the most prominent data repository solutions.

1 http://infolab.fe.up.pt/

- **José Devezas** (InfoLab) whose skill with recommender systems and overall interest in technology always made me think of different ways to solve problems.
- **Melinda 'Meli' Oroszlányová** (InfoLab) and **Pedro Brandão Silva** (GIG @ FEUP[2]), for their interesting questions and constructive criticism which contributed to the definition of the mathematical formula for the descriptor ranking algorithm. Meli, you are also the sweetest person to ever grace InfoLab, and a very much missed element of the InfoLab swimming team.
- **Manika Kar** (InfoLab), who helped me to use the R statistical analysis platform while burning my palate with her super tasty but also—at first—super spicy Indian cuisine.
- **João Jacob** (GIG @ FEUP), who is always willing to exchange constructive criticism on any technical solution and is simply a top-notch person.
- All the others that I forgot to mention.

I also give a special thanks to my friends who encouraged me through this work, despite the many times I was absent because of it. Last but definitely not least, I would like to thank my family, who always supported me unconditionally.

SUPPORT FUNDING ACKNOWLEDGMENTS

This work's main funding support was given by PhD grant **SFRH / BD / 77092 / 2011** provided by the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

This work is supported by project "SIBILA" (**NORTE-07-0124-FEDER-000059**), financed by the North Portugal Regional Operational Programme (ON.2-o Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF) and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

2 http://paginas.fe.up.pt/~gig/

Simplicity is the ultimate sophistication.

LEONARDO DA VINCI

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACID	Atomicity, Consistency, Isolation, Durability is a set of properties that guarantee that database transactions are processed reliably [89]. 81, 106
Application Profile	An Application Profile is defined by the Dublin Core Metadata Initiative (DCMI) as a declaration of which metadata terms an organization, information resource, application, or user community uses in its metadata [52]. 14, 19
BLAST	BLAST is an algorithm designed to calculate similarities for biological sequences, used in genetics and biology research. More information can be found in the original publication [9]. 48
Content Management System	A Content Management System is a computer application designed to allow the creation, editing, publishing, modification and maintenance of content from a central interface, as well as its related metadata [146]. 33
Content negotiation	HTTP Content negotiation allows, among other things, a client to request information from a server in a specific format, if available, using only features present in the HTTP protocol itself. According to the RFC 7231, section 3.4 [4]: "when responses convey payload information, whether indicating a success or an error, the origin server often has different ways of representing that information; for example, in different formats, languages, or encodings. Likewise, different users or user agents might have differing capabilities, characteristics, or preferences that could influence which representation, among those available, would be best to deliver. For this reason, HTTP provides mechanisms for content negotiation.". 81
Controlled Vocabulary	A controlled vocabulary is an established list of standardized terminology for use in indexing and retrieval of information. An example of a controlled vocabulary is subject headings used to describe library resources [157]. 14

Data curator
: A data curator is a person responsible for performing data curation, which is "the activity of, managing and promoting the use of data from its point of creation, to ensure it is fit for contemporary purpose, and available for discovery and re-use. For dynamic datasets this may mean continuous enrichment or updating to keep it fit for purpose. Higher levels of curation will also involve maintaining links with annotation and with other published materials"[138] . 3, 96

Foreign Key
: In relational databases, a foreign key is a field, or a collection of fields in one table that uniquely identifies a row in another table [56] . 100, 106

Nepomuk File Ontology
: An ontology designed for "describing native resources available on the desktop". See http://www.semanticdesktop.org/ontologies/2007/01/19/nie/ . 102

Ontology
: Ontologies are, in short, "specifications of conceptualizations" [87]. Ontologies are closely related to the semantic web [104] and are excellent tools for defining relevant concepts of a domain (the types of things and relationships between those things). They also provide higher-level representations of datasets because they do not rely on the structure or document syntax features such as nesting or numbering of elements [113]. They not only provide richer semantic representations but can also be used to infer knowledge over existing sets of relationships between different entities, something that can play an important role in the retrieval of research information ("semantic search"). Ontologies evolve through their reuse, making it one of the goals to aim for when designing one [35]. The common understandings that arise from that reuse are the basis for interoperability [14]. 15, 81

Open Data
: Open Data refers to Data available under open conditions. According to the Open Knowledge Foundation (OKF), the concept of Open, when applied to a knowledge work, means that "The work shall be available as a whole and at no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The work must also be available in a convenient and modifiable form." [150]. 1, 34, 37, 46

Open Government
: Open government is the governing doctrine which holds that citizens have the right to access the documents and proceedings of the government to allow for effective public oversight [130]. 47, 48

Persistent Identifier — A persistent identifier "is an association between a character string and an object. Objects can be files, Parts of files, persons, organizations, abstractions, etc. The identifiers can be Uniform Resource Names (URNs), Digital Object Identifiers (DOIs), or Uniform Resource Identifiers (URIs)" [53]. 4, 32

Representation Information — Representation information is, according to the OAIS recommendations [59], additional information that accompanies a physical or digital object, and that is necessary for expressing its meaning. For physical objects, it can record physically observable attributes of the object. For digital objects, it is necessary to adequately map the sequences of bytes or characters that make up the digital object into their higher-level meanings. As such, representation information is that information necessary to correctly interpret a sequence of bits as a usable digital object, such as a document or a file. 11, 15, 28

Triple Store — A triple store is designed to store and retrieve identities that are constructed from triplex collections of strings (sequences of letters). These triplex collections represent a subject-predicate-object relationship that more or less corresponds to the definition put forth by the RDF standard[3] . 81

3 http://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/rusher.html

ACRONYMS

ACM	Association for Computing Machinery. 71
AIP	Archival Information Package. 29, 37
AP	Application Profile. 18–20, 24, 26, 50, 54, 85, 93, 96, 100, 163, 165, 168
API	Application Programming Interface. 6, 18, 22, 23, 29, 30, 32, 35, 37, 38, 41, 45, 46, 49–51, 54, 82, 84, 96, 100, 104, 105, 107, 168, 169
CERIF	Common European Research Information Format. 51, 112, 119
CERN	Conseil Européen pour la Recherche Nucléaire. 46
CIFS	Common Internet File System. 43
CKAN	Comprehensive Knowledge Archive Network. 18, 33, 36–42, 49, 51, 54, 77, 78, 82, 84, 86, 89, 100
CRIS	Current Research Information Systems. 51, 54
DAF	Data Asset Framework. 30
DANS	Data Archiving and Networked Services. 46
DC	Dublin Core. 12–14, 16–18, 24, 31, 40, 42, 44, 48, 78, 111, 114, 116, 125, 127, 128, 131, 132
DCAP	Dublin Core Application Profile. 7, 85, 100
DCB	Double Cantilever Beam. 83, 92, 101, 114, 141
DCMES	Dublin Core Metadata Element Set. 12, 14, 15, 125, 126
DCMI	Dublin Core Metadata Initiative. 7, 13–15, 19
DCTERMS	DCMI Metadata Terms. 12, 15, 54, 79, 83, 125, 126, 132, 141, 150, 168
DDI	Data Documentation Initiative. 13, 14, 16, 47, 111
DIDL	Data Item Declaration Language. 40
DIP	Dissemination Information Package. 28, 29, 37, 40
DMP	Data Management Plan. 43
DOI	Digital Object Identifier. 32, 44
DTD	Document Type Definition. 16, 50
EML	Ecological Metadata Language [142]. 111
EPSRC	Engineering and Physical Sciences Research Council. 1
EU	European Union. 7, 46, 51, 115
FEUP	Faculdade de Engenharia da Universidade do Porto. 141
FOAF	Friend Of A Friend Ontology. 15, 92, 111, 116, 119, 141, 150, 153
GBIF	Global Biodiversity Information Facility. 41

HITS	Hyperlink-Induced Topic Search. 60
HTML	Hypertext Markup Language. 81, 105
HTTP	Hyper Text Transfer Protocol. 22, 105, 107
ICPSR	Inter-university Consortium for Political and Social Research. 3, 12, 16, 47, 79
IEEE	Institute of Electrical and Electronics Engineers. 17, 111
IPR	Intellectual Property Rights. 13, 17, 28
IR	Information Retrieval. 58, 59, 61, 64, 71, 73, 144
ISO	Industry Standards Organization. 14
JISC	Joint Information Systems Committee. 1, 46
JISC-UK	Joint Information Systems Committee UK. 48
JSON	JavaScript Object Notation. 81, 95, 98, 105, 107, 121
LDA	Latent Dirichlet Allocation. 69
LDSD	Linked Data Semantic Distance. 68
LINQ	Language Integrated Query. 104
LOC	Library of Congress. 16
LOCKSS	Lots of Copies Keep Stuff Safe. 168
LOD	Linked Open Data. 17, 21, 24, 30–32, 45, 46, 54, 57, 68, 69, 80, 81, 83, 96, 106, 110
LOM	Learning Object Metadata. 17, 20
LOV	Linked Open Vocabularies. 21, 22
MAE	Mean Average Error. 61, 64
MARC	Machine Readable Cataloguing. 14, 16, 17, 40, 42, 48
METS	Metadata Encoding and Transmission Standard. 14, 16, 40
MODS	Metadata Object Description Schema. 16, 17, 40
NCBI	National Centre for Biotechnology Information. 3, 12, 48
NIH	National Institutes of Health. 4
NISO	National Information Standards Organization. 13
NSF	National Science Foundation. 4
OAI-PMH	Open Access Initiative—Protocol for Metadata Harvesting. 18, 30–32, 37, 38, 40–42, 44, 45, 50, 51, 54, 96
OAIS	Open Archival Information System. 28–30, 34, 37
ODBC	Open Database Connectivity. 107
OKF	Open Knowledge Foundation. 49, 89
ORM	Object-Relational Mapping. 104, 107
OWL	Web Ontology Language. 7, 18, 20, 21, 81, 82, 105, 110, 111, 116, 121, 168
PDF	Portable Document Format. 98, 134

PREMIS	PREservation Metadata: Implementation Strategies. 16
RDA	Research Data Alliance. 50
RDF	Resource Description Framework. 7, 12, 15, 17, 20, 22, 24, 54, 81–83, 97, 98, 104, 105, 110, 121, 125
RDM	Research Data Management. 1, 6, 11, 18, 24, 43, 46, 54, 77, 146, 162, 169, 170
RE3Data	Registry of Research Data Repositories. 50
REST	Representational State Transfer. 6, 50, 96, 104
SFTP	Secure File Transfer Protocol. 43
SIP	Submission Information Package. 29, 37
SPARQL	SPARQL Protocol and RDF Query Language. 21, 31, 81, 84, 88, 93, 101, 102, 104, 106, 110, 119, 121
SPSS	IBM SPSS Statistics Software. 48
SQL	Structured Query Language. 45
SSH	Secure Shell. 43
SWM	Semantic MediaWiki. 79, 91–93
SWORD	Simple Web-service Offering Repository Deposit. 42, 44
UI	User Interface. 21, 72, 73, 125, 126
URI	Uniform Resource Identifier. 12, 15, 19, 23, 31, 32, 96, 101, 109, 125
VoID	Vocabulary for Interlinked Datasets. 21, 112
W3C	World Wide Web Consortium. 22
WebDAV	Web-based Distributed Authoring and Versioning. 43
XML	eXtensible Markup Language. 13, 14, 16, 17, 19, 20, 24, 40, 42, 47, 48, 50, 51, 81
XSD	XML Schema Definition. 14

1 INTRODUCTION

Research data management is a problem with specific issues that range from the political and ethical issues concerning data sharing to the technical aspects of how to make data reuse possible. The debate on Open Data is complex and closely related to this work. However, we do not discuss whether or not data should be open; instead, we assume that such policies are in place and focus on improving those workflows from a technical standpoint.

The number of published scholarly papers is steadily increasing, and there is a growing awareness of the importance, diversity and complexity of data generated in research contexts [148]. The management of these assets is currently a concern for both researchers and institutions who have to streamline scholarly communication, while keeping record of research contributions and ensuring the correct licensing of their contents [140, 95]. At the same time, academic institutions have new mandates for the management of their research data [72, 154]. The data management plans that are are now becoming mandatory parts of research grant proposals, outlining the data management activities to be carried out during the research projects. Such activities are invariably supported by software platforms, increasing the demand for these infrastructures.

Research data is among the most valuable kinds of data, since research datasets may be useful for entire research communities and also because research findings can have an impact on society as a whole [29]. Throughout this work we adopt the definition of research data as presented by the Engineering and Physical Sciences Research Council (EPSRC) and adopted by the Joint Information Systems Committee (JISC) in its recommendations for Research Data Management (RDM) in UK Universities [41]: "recorded factual material commonly retained and accepted in the scientific community as necessary to validate research findings" [71].

The emerging "Fourth Paradigm of Science", which comes from the increasingly widespread access to powerful computing capabilities in networked environments [100] is now allowing researchers to manipulate and produce larger and larger datasets [99]. As a consequence, research data management is one of the greatest challenges of the information age, and a much needed response to the "data deluge" [139]. Proper RDM ensures the reproducibility of research findings, enables the placement of new research questions and increases the visibility of researchers and institutions. Data publication is growing quickly in the recent trend of data-intensive science and, unlike traditional academic papers, datasets raise distinct concerns when it comes to their publication. On the one hand, data have been regarded in many areas as valuable assets which should be explored but not disseminated, to secure future uses. On the other hand, the publication of many datasets is hindered by intrinsic obstacles such as confidentiality and business contracts [36].

1.1 RESEARCH DATA MANAGEMENT IN THE LONG TAIL

In 2011, Science carried out a survey over 1700 of the magazine's peer reviewers, concerning the availability and reuse of research data [201]. It was concluded that more than half of them archive their data within their laboratories, a situation that is probably caused by lack of proper support for data curation, since a vast majority (more than 80%) considered that their funding for data curation was insufficient. The survey also concluded that more than half "rarely" reused base data from the published literature for their original research papers—however, it is interesting to note that more than 60% of the polled individuals directly contact the authors of papers that they find in order to obtain the base data. It is therefore reasonable to conclude that: 1. There is a need for base data to be linked to research papers, as the data requests highlight and 2. Re-users contact data creators to get data *that they can understand*; contacting the data creator is a way to get not only the files themselves but also the information about the data (metadata) that is necessary to understand it.

From the 1700 respondents, almost half stated that the size of their datasets was less than 1GB, showing that there is much room for improvement in research data management without delving into "big data territory". More than 80% also stated that there was no funding available in their research group for data curation. This problem is aggravated in the current context of massive data creation, particularly in research groups with access to limited resources [37].

These traits are a part of the so-called "Long Tail of Science" [95], the multitude of small research groups that produce small volumes of data in terms of storage, but that together make up for a huge number of potentially valuable but often neglected datasets. Recently, a shift has been pointed out in this definition, as the democratization of devices capable of producing large quantities of data has made it possible for small research groups to produce huge volumes of data [172]. As cloud computing and storage become more accessible, data management solutions targeting the long-tail should be able to scale and deal with such storage demands. The contributions of this thesis work and dissertation apply mainly to these research groups in the long-tail of science, helping them manage their data throughout the research workflow.

The survey also provides interesting insight on the data sharing practices of the respondents. An overwhelming majority (more than 75%) had asked colleagues for base data related to their published papers, and only 3,7% of those requests were not met with positive responses. This is a clear indication that 1. researchers express a very strong interest in obtaining base data for the papers that they read and 2. researchers are willing to share the data they create. Thus, researchers show availability to provide data and spend some of their time to explain it to their peers. Our experience with a panel of researchers confirms this, since researchers tend to contact their peers and ask for datasets, but not just the data. By contacting the data creator, they can inquire directly about the dataset's production context, and are more confident that this metadata will be accurate and adequate for them to decide whether or not to reuse the data and how to do so. In alternative, they could provide their datasets to the public and associate metadata records to them, so that other researchers would not need to contact them directly. In practice, this happens rather rarely, since the effort involved in producing

metadata records rich enough for public sharing is much higher than that required for personal cataloguing and sporadic sharing [36]. Past work has showed, however, that researchers themselves feel the need to produce basic metadata records at an initial stage of the research workflow, even if only for their personal management of the datasets that they produce [114, 193].

1.2 MOTIVATION

The success of a data management initiative can be measured by the ability of third parties to reuse the preserved research products. However, to achieve the goal of data reuse, high-quality metadata must be present [138], and the specific nature of research datasets often requires researchers to work together with data curators to adequately capture the production context of a dataset [141]. In reality, however, curators are often unavailable in many research groups, especially in those groups with limited financial resources to ensure proper curation of their research products—the so-called long-tail of science [95]. As a result, researchers themselves often have to perform description tasks on their data, either by their own initiative or as a result of funding requirements [131]. They are not experts in data management nor metadata, so they need adequate and unobtrusive tools to assist them in the process, that should support them while placing the minimum possible overhead on their research work. In some domains, like biology or social sciences, there are established practices, technology and platforms for data sharing (the National Centre for Biotechnology Information (NCBI)[1] and the Inter-university Consortium for Political and Social Research (ICPSR)[2] are good examples). Groups in the long-tail of science, however, need to reduce their dependence on manual curation, by being provided with a good "staging area" for data description that can allow their datasets to reach the long-term preservation phase.

Studies have demonstrated that papers that provide base data show consistently higher citation rates over time, providing additional recognition of the author's work through data citation [171, 170]. With *informal* data sharing taking place even despite the lack of adequate resources for data curation [201], it is clear that many researchers are willing to share their data, and that there is a great demand for datasets with high-quality metadata as a complement of the original publications. On the other hand, research data description is perceived as an expensive process in terms of time and resources.

Data publication also plays a very important role in ensuring the reproducibility of research findings [221]. Journal articles are becoming increasingly driven by data, with the number of journals that have published at least one *Data Paper* steadily increasing. The very definition of Data Paper is defined as having "at least two elements that have to be materialized into concrete and identifiable information objects, a dataset and a paper that describes that dataset" [45]. Data papers can be seen as a temporary solution to enable dataset citation, at least until the current culture of dataset citation evolves to the same level as the practice currently present in the research community: it is hard to get citations for datasets, even when they are under repository control and possess unique identifiers—but when there is a paper that describes a dataset and directly points to it, that paper can itself

1 http://www.ncbi.nlm.nih.gov
2 https://www.icpsr.umich.edu/icpsrweb/landing.jsp

serve as the citable object. Thus, one can say that the reasons for the lack of dataset citations are still primarily cultural instead of technical.

Research funding institutions also play a very important role in the implementation of adequate data management and publishing policies. While some years ago data publishing and reuse was still in its infancy [155], great progress has been achieved since. The European Union has introduced new guidelines on data management for the Horizon 2020 calls [73], while in the US, funding agencies such as the National Science Foundation (NSF) or the National Institutes of Health (NIH), have mandatory data management plans [154, 80].

1.3 GOALS AND CONTRIBUTIONS

The dataset management lifecycle can be divided into two separate moments: properly gathering and describing the data, making sure that it reaches a long-term repository, and the preservation processes required to ensure that the data can be reused in the long-term. We focus on the first part, making sure that those datasets reach the late stages of the preservation workflow. To achieve this, we propose to make data description more agile by assisting in the selection of those descriptors that are more suited to each research domain or research project.

We want to make it easier for researchers to describe their data in a more autonomous manner, reducing the dependency on curators throughout the research workflow. This would allow curators to assume more the role of metadata validators than that of metadata producers. At the same time, a quicker description and deposit would allow—in an ideal scenario—the researchers to cite datasets that support their publications at the time of submission, even before they are actually published. Access restrictions would still apply, of course, but the embargo conditions would be handled by the long-term data repository. The minting of a Persistent Identifier for the new data (that can be used to cite it) would be also carried out by that platform.

Thesis Statement. *Usage information can be used to build application profiles for the description of research data assets, in a semi-automated manner, reducing the effort required to produce high-quality, domain-specific metadata records.*

One of the main purposes of this work is to prove this hypothesis. We argue that it is possible to take advantage of descriptor usage information to measure the preference of users for certain sets of descriptors, in order to design custom application profile for each research domain, and in a partially automated way.

Research Question 1. *Can a collaborative research data management environment engage researchers in the management of their own research data?*

Research Question 2. *Can usage information be used to provide adequate recommendations of descriptors for distinct research domains?*

Research Question 3. *Can researchers, when supported by collaborative research data management tools, produce metadata records that are satisfactory as judged by a curator?*

These research questions are a separation of the hypothesis into subproblems. Research question 1 aims to determine if researchers are willing

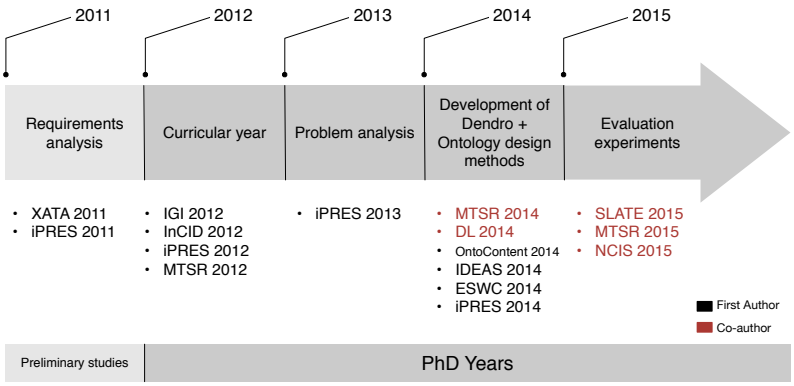


Figure 1: Timeline of contributions yielded by this work

to carry out a collaborative description effort over their own data, which is what we want to foster with this work. Research question 2 covers the means through which we prove our hypothesis: a descriptor recommendation system supported on actual usage information. Finally, research question 3 aimed to determine if the quality of the metadata records remains satisfactory after the introduction of the recommendation approach proposed in our work.

Figure 1 shows the publications written along the timeframe of this work. The design of a research data management workflow should start with requirements gathering, where current data management processes are analysed and mapped. In 2011[3] we carried out such a study with the collaboration of a panel of 13 researchers and research groups from the University of Porto, whose research domains included Chemical Engineering, Mechanical Engineering, Civil Engineering, Biology, Gravimetry, Education Sciences, Psychology, Ecology, Economics, and Civil Engineering. This initial study yielded an overview of the data management needs of the interviewed researchers, which was then used to select several features to be implemented in a research data repository [196]. This prototype, built on the DSpace publication repository[4], was deployed at U.Porto and tested with the collaboration of the panel [49].

The first steps towards answering research question 2 were taken in 2012. Our first approach at descriptor recommendations was implemented as a prototype that takes advantage of information from DBpedia[5] to suggest possible properties for documents by analyzing their contents. In this case, the triple-based prototype uses Apache Jena and Lucene to perform textual analysis on the contents of a file; a set of *top terms* were extracted through the $\mathtt{tf} \cdot \mathtt{idf}$ measure [143]. These terms are then used as queries over DBPedia, retrieving a list of DBPedia resources; the outgoing edges of these resources (instances of properties that have those resources as their subjects) are counted and grouped. The top-N ranked properties are then be recommended as possible descriptors for the document initially provided [194].

Answering research question 1 meant putting a collaborative RDM system in place, either through reuse or development from scratch. Steps towards a semantics-based integrated workflow for data deposit and metadata representation were then taken in 2013. They were informed by the

3 This work was part of a 1-year preparation for this PhD, supported by U.Porto, Universidade Digital.

4 Available at http://www.dspace.org

5 http://www.dbpedia.org

requirements gathered in 2011, and drew from the lessons learned from the DSpace developments of 2011. Now targeting the early description of research datasets, we developed the UPBox and DataNotes prototypes [195, 188]. A first approach towards answering research question, these two solutions were put in place and tested with researchers [49]. The UPBox solution was designed as a cloud-based file storage platform (similar to Dropbox[6]), with a full Representational State Transfer (REST) Application Programming Interface (API) to enable its integration with other modules. It also included an easy-to-use web interface, and was intended to provide a safe and private storage platform for small research groups. UPBox was intended to complement DataNotes, a data description solution built on Semantic MediaWiki[7]. These developments, while limited in extent, showed the direction that the work should take: a fully triple-based, cloud-ready storage solution with wiki-like features, capable of supporting collaborative dataset production and semantic metadata production—all while maintaining an easy-to-use interface that researchers could use autonomously.

Drawing from the experience gathered during the implementation of UPBox and DataNotes, a new platform was designed from scratch. Named *Dendro*, it is a fully triple-based dataset and collaborative metadata editing [190, 189]. It combines those features with a file storage layer designed with scalability in mind, so that it can handle the large numbers of files that researchers can create, even in small research groups. Dendro is integrated with many known open-source repository platforms for data publishing. The platform's development was guided by the requirements of users from the start; perhaps the most important were ease of use, and the ability of users to centralize their data and share it with their colleagues in a controlled manner. Thus, we drew inspiration from existing user-friendly file management solutions such as Dropbox.

The rich API of Dendro allows its integration with other solutions, and an example of such integration is *LabTablet*. It is an Android-based electronic laboratory notebook that allows the collection of multiple types of metadata using the onboard sensors of a mobile device (luminosity, geolocation, temperature, camera) and their deposit in Dendro as instances of properties from different ontologies [10].

The data model of Dendro is fully triple-based, with no relational database at the core of the data layer. The advantages of a triple-based database were highlighted, in the context of this particular problem of data management. This type of database makes it easier to model requirements usually hard to represent in a relational model—file and folder hierarchies, unknown attributes for unknown entity types at the time of modeling, and resource versioning [197]. Another important requirement for Dendro is its ability to represent domain-specific metadata for the many possible research domains. Unlike relational models, that have their structures fixed at the time of modeling, the triple-based model is more flexible, since it allows a finer representation of resources of different types.

Dendro's graph-based data model is able to grow without the need to modify the source code of the platform, by loading additional sets of metadata descriptors. These sets of descriptors are formalised as *lightweight ontologies* (modeled as Web Ontology Language (OWL) or Resource Description Framework (RDF) in an ontology editor such as Protégé) that could be loaded into the data model of a Dendro instance, making their descrip-

6 http://www.dropbox.com
7 https://semantic-mediawiki.org/

tors available for use by the researchers. This approach for representing metadata schemas as ontologies was first presented using two Engineering domains (Mechanical and Analytical Chemistry) [48] and further tested using it in the Ecology domain. This method was further instantiated through the modeling of an ontology for describing geo-referenced ecology occurrences [191], based on the INSPIRE European Union (EU) directive for representing geospatial metadata [25].

To answer research question 2 we performed two experiments where Dendro was progressively used as the basis for providing descriptor recommendations suited to each researcher, and indirectly to each research domain. By gathering data about the behaviour of users during the description of their datasets (which descriptors are used, hidden or *favorited*), we were able to present the descriptors that users were more likely to fill in, and hide those that were not interesting. After this experiment, we concluded that domain-specific descriptors were more easily found and used when the recommendation was in place, and also that the users had to spend *less effort* to fill in a similar number of descriptors.

The deposit of research publications in a repository can, in most cases, be carried out by data management professionals in an autonomous fashion, since they are proficient at metadata production and are used to handling complex self-deposit workflows. The publication of research data, on the other hand, has to start from the researchers, since they know the semantics of the very specific metadata that needs to be added to the dataset to enable its later reuse. Since these users are not data management experts, user interfaces need to show just enough information to allow them to fill in the adequate metadata, without burdening them with the metadata idiosyncrasies often present in self-deposit workflows. This led us to formulate the research question 3, which aims to draw conclusions about the quality of the metadata records produced by the researchers. Our experiments took this into account by incorporating evaluations by a data curator as well as by the senior elements of research groups.

Lastly we argue that in a data publishing scenario, actual usage data can be a valuable contribution towards the production of domain-specific metadata standards. Broadening this approach to several research institutions with similar domains could provide real usage data that could foster the agreements between those groups, and perhaps drive the creation of Dublin Core Application Profiles (DCAPs). This way, even research groups that lack the curatorial resources to follow established guidelines—such as those proposed by the Dublin Core Metadata Initiative (DCMI) for the creation of DCAPs [60]—could contribute with their usage statistics to provide evidence as to which descriptors should actually be included in a DCAP suited for a specific domain.

Several masters' thesis stemmed from the work on this Ph.D. They are listed as follows:

- "Pinho, M. (2012). Modelo de Replicação para a Preservação de Dados em Repositórios" [168]
- "Gouveia, M. (2013). DataNotes — Um sistema colaborativo para anotação de estruturas de diretórios" [83]
- "Barbosa, P. (2013). UPBox : Armazenamento na Nuvem para Dados de Investigação da U . Porto" [24]

- "Castro, J.. (2013). Estudo de utilização do Repositório de dados da Universidade do Porto" [49]
- "Amorim, R. (2014). LabTablet: A multi domain laboratory book" [11]

1.4 DISSERTATION STRUCTURE

This dissertation is organized as follows: Part I provides an overview on research data management practices, initiatives and projects. Part II covers recommendation techniques, recommender systems and evaluation metrics and procedures. Part III presents and documents Dendro, the research data management prototype platform that we built to test and prove the presented hypothesis. Part IV shows the recommendation approach designed to help researchers select the most adequate descriptors for their datasets, as well as its implementation in the Dendro platform. Finally, Part V presents some conclusions that can be drawn from this work and outlines some research lines for future work.

Part I

Research data management

2 METADATA

In order to provide a brief overview on the issue of metadata for research data management, this chapter starts by introducing the concept, how it is usually derived, and the common problems faced by communities of different sizes and domains when producing and managing their metadata records. In this overview, we do not place a strong focus on preservation metadata and its supporting models, as we consider this topic out of scope of this work. The metadata that we are trying to gather here is not related to preservation processes but rather regarding experiments, methods or other concepts closer to researchers. We consider that preservation metadata should be left for the curators to handle, as they are the experts in such domains.

The importance of metadata for the management of research data is discussed, with the introduction of the metadata schema as a way to ensure the structural correctness of metadata records. Some of the most prominent metadata schemas are presented and compared, and the concept of *Application Profile* is introduced as a way to tailor metadata to the needs of a particular community.

Given the increasing importance of LOD in the context of Research Data Management (RDM) initiatives, we cover the differences between schema-compliant metadata records and semantic metadata supported by ontologies. Ontologies are building blocks of the semantic web, but it can be hard to find an ontology that defines the relevant concepts for a domain of application. Repositories of ontologies play a very important role in dealing with these issues, so we list a few of them and cite some of their most important features.

2.1 INTRODUCTION

An essential part of data curation is capturing a dataset's Representation Information, which is the term used to designate all the relevant information that is required for a researcher to correctly interpret and reuse a dataset [51]. Capturing the semantics of research data is especially difficult because of its inherent complexity and specifics—representation information may use domain-specific terms which can only be correctly understood by researchers working on that dataset's particular research domain.

Representation information is usually represented by metadata ("data about data") that can be formalized in the form of controlled vocabularies which are "established lists of standardized terminology for use in indexing and retrieval of information. An example of a controlled vocabulary is subject headings used to describe library resources" [157]. Metadata can also be generic—author, creation date and location, associated publications—or specific—in the case of a chemistry dataset, a metadata element may refer to the chemical agents used for the experiment that yielded the data. Dublin

Core (DC) is a widely used generic metadata profile for describing resources [111].

The presence of high-quality metadata associated to research data assets is an essential step for their re-use and preservation [138]. The production of good metadata records is an expensive process, since it requires some knowledge of information management. Given the often specific nature of the resources being described, the collaboration of the data creators with data scientists (also called curators) has been identified as an important step towards the production of metadata records that adequately capture a dataset's production context [141]. Usually, the original depositor of the data fills in a simplified metadata sheet, that is later translated into a structured representation by a curator [21]; this representation is often made using existing metadata schemas, from which the Dublin Core Metadata Element Set (DCMES) schema[1] is arguably the most widely used due to its simplicity and maturity.

It is clear that metadata production for research datasets is very different when compared to research publications. It must include the participation of the researchers, as only they possess the knowledge of the domain that is required to adequately document their datasets' production context so that others can reuse them. The challenge often is, however, to encourage them to participate in deposit stage, as the investment placed in metadata production for sharing with the broad public is often much higher than the implied in the production of notes that are only to be reused by the creator or within the research group [36].

While the basic DC schema has served the community for a long time, it was developed before the invention of Resource Description Framework (RDF), and therefore before the RDF notion of *range* was matured into the latter. This introduced some ambiguity in the specification: should the string containing the name of the creator of a resource be a valid value for an instance of a dc:creator descriptor, or should the Uniform Resource Identifier (URI) that identifies the author be considered only?[2].

An extended DC specification—DCMI Metadata Terms (DCTERMS)—was implemented in 2008. It provided a set of 15 descriptors, specified as subproperties of the original DC Element descriptors (not to "break" compatibility with existing records), and complemented it with several others designed to reduce ambiguity and improve the expressiveness and specificity of the metadata values. The newer schema provides additional ways to convey relevant metadata, but at the same time adds increases complexity, since not all elements are relevant for every type of resource created during research efforts.

While it would be desirable to obtain metadata records as comprehensive as possible for every research asset, it is unreasonable to believe that researchers—who are not data management experts and whose time is too valuable—would spend a lot of time filling in all of the metadata descriptors for every resource they produce. In some domains, like biology or social sciences, there are established practices, technology and platforms for data sharing (National Centre for Biotechnology Information (NCBI)[3] and Inter-university Consortium for Political and Social Research (ICPSR)[4] are good respective examples). Many other research domains, located in the

1 http://dublincore.org/documents/dces/
2 Example drawn from the Dublin Core documentation wiki http://wiki.dublincore.org/index.php/FAQ/DC_and_DCTERMS_Namespaces
3 http://www.ncbi.nlm.nih.gov
4 https://www.icpsr.umich.edu/icpsrweb/landing.jsp

so-called "long-tail of science", those groups with insufficient resources to handle data curation [95], need tools capable of reducing their dependence on manual curation, while still providing them with the basic means to place their datasets into repositories.

In this chapter we will cover both semantics and syntax for metadata representation. In terms of semantics, we will discuss some metadata schemas and recommendations for the representation of metadata records, such as DC or Data Documentation Initiative (DDI). In terms of syntax, we will analyse several alternatives for the representation of metadata, such as eXtensible Markup Language (XML). Their applications in some data management projects will serve as the basis for discussing their advantages and disadvantages in the representation of metadata records.

2.2 METADATA AND SEMANTICS

Metadata representation relies on shared semantics at all levels, since systems need to agree on the meaning of metadata records in order for them to be correctly interpreted outside of the system where they are created. The need for interoperability has driven metadata groups such as the Dublin Core Metadata Initiative (DCMI) to establish standards for metadata representation. These standards are important for all stakeholders of the data management ecosystem, helping ensure interoperability and also that metadata records are comprehensive and understandable.

2.2.1 Types of metadata

The National Information Standards Organization (NISO) divides metadata into three main categories [153]:

- **Descriptive**
  Descriptive metadata describes a resource for purposes of discovery and identification. Examples are the author, title or keywords.
- **Structural**
  Structural metadata describes how the different parts of the resource being described are compound together. For example, the chapter numbers in this dissertation explain how they should be ordered in the structure of the document.
- **Administrative**
  Administrative metadata includes all that information required to manage a resource, including how it was created (for example, methodology or software or other technical information) or who can access it. There are two subcategories within the scope of administrative metadata:
  - **Rights management metadata**
    Concerned with documenting Intellectual Property Rights (IPR) information, and
  - **Preservation metadata**
    Information necessary to preserve a resource. For example, the format in which the resource is encoded (if applicable), intervention of migration schedules or a detailed account on the full software stack required to access that resource.

2.2.2 Metadata schemas

A metadata schema defines a set of rules for representing metadata. A schema usually includes both syntactic and semantic definitions of the terms used in the schema. This means that, not only does it define how a schema-compliant record must be structured, but also establishes the meaning of each term and of the relationships between different terms in the same schema. Such structure constraints are usually formalised using XML technologies such as XML Schema Definition (XSD) [152].

Metadata schemas can be generic or tailored to a specific domain. Generic metadata schemas have been designed for bibliographic purposes [88] and are more easily understandable by individuals without knowledge of a particular research domain. For resources of specific nature like research datasets, however, they often need to be refined to satisfy the needs of each community, leading to the creation of many different metadata schemas [18].

A metadata schema is usually composed of the following elements:

- A set of terms that are specified by the schema designer. In the case of DC, for example, the schema specifies elements such as Title, Author or Date Created.
- A set of terms that constrain the possible values and/or terms that can be used as values for the term instances, or property values. These can resort to Controlled Vocabularies, such as the DCMES, [5] the ISO-8601[6].
- A set of rules for the structural constraints and syntactic features that are to be present independently of the implementation of the schema, namely the mandatory nature of some terms, their cardinality, and others. These rules are also present in Application Profiles.
- A set of binding rules for implementing the schema in a specific description language, such as XML.

Some examples of metadata schemas used in dataset description are DC, DDI, Machine Readable Cataloguing (MARC) and, at a different level, Metadata Encoding and Transmission Standard (METS)—we will cover these schemas summarily.

Dublin Core

In 1998, the DCMI created DC, a generic metadata schema that is easy to understand even by non data management experts. DC is perhaps the most widely used metadata schema, and was designed as a generic metadata schema to describe practically any resource available on the web.

In its simplest form of 15 descriptors, called DCMES, it is easy to understand and implement, even by non data management experts. This version includes descriptors such as Title (a name given to a resource), Contributor (an entity responsible for making contributions to the resource) or Date (a point or period of time associated with an event in the lifecycle of the resource)[7]. The apparent simplicity of this schema is its greatest strength,

5 The DC schema that specifies the base 15 base elements. See http://dublincore.org/documents/dces/

6 The Industry Standards Organization (ISO)-standard that covers the representation of date values

7 Information available at the DCMI website for the DCMES specification: http://dublincore.org/documents/dces/

but also arguably its greatest weakness; while simple to understand at first, some descriptors are ambiguous in their semantics and the metadata that can be represented using this schema is limited. For research datasets, which is our domain of study, such ambiguity and lack of detail makes the DCMES a incomplete alternative for the capture of the Representation Information of a dataset.

To deal with the lack of detail of the DCMES, the DCMI have created an improved version, called DCTERMS. The DCTERMS schema emerges after the creation of RDF, and as such, uses the notion of range provided by the latter. The advantage of specifying the range of a descriptor is that, unlike DCMES, DCTERMS specifies the types of acceptable values for each descriptor, at a semantic level. Figure 2 illustrates this difference. It shows that, while in the DCMES the string representation of a Contributor descriptor could contain the full name of the *Agent* that created a resource and be considered valid, the DCTERMS specifies that the Contributor of a resource must be an instance of the http://purl.org/dc/terms/Agent class. Thus, if the same string containing the full name of the person is used as the Contributor of a DCTERMS metadata record, it would not be correct. The character string must instead represent the URI of a resource that represents the Agent in question, who in turn would have its own name, represented as an instance of the Friend Of A Friend Ontology (FOAF) ontology properties firstname and surname, for example. This helps solve the problem of ambiguity of the specified metadata values by using URIs to uniquely identify the related resources, instead of relying on ambiguous string representations (in this case, there can be many people with the name "João Rocha", but only a single URI can identify the person in question).

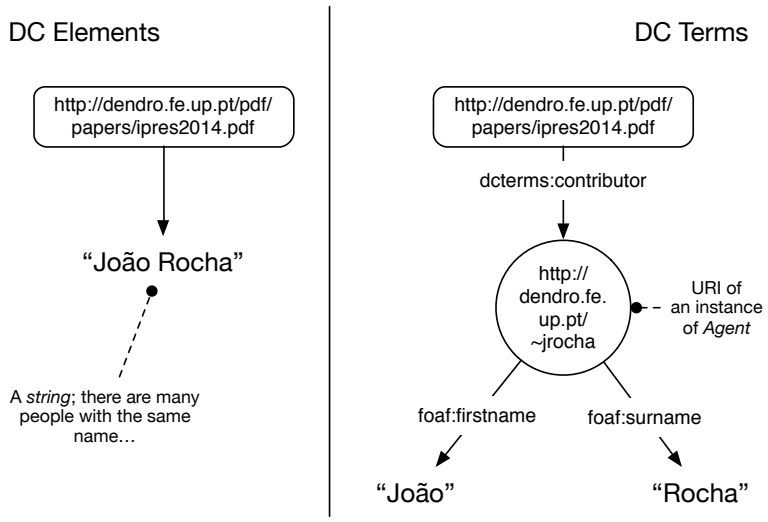


Figure 2: Representing metadata using the DC Elements and DC Terms schemas

The downside of DCTERMS representation is that it introduces an additional layer of complexity to the metadata model, and requires some knowledge of the concepts of *Linked Data* as well as of Ontologies. This makes the schema harder to understand by data curators and very difficult to comprehend for researchers, who are not accustomed to the notions of *resource* or *property*.

DDI

The DDI is a large international project which has the goal of standardizing metadata records for statistical datasets, commonly present in the social and behavioural sciences. It is used by ICPSR, a very prominent repository for social science datasets, to represent its metadata records [153].

An interesting aspect of the DDI is that it allows for the hierarchical description of social science studies, including the files that result from it as well as the variables measured or referenced in the study. The ICPSR also emphasises the description of the variables present in a dataset in the portal, as there can be thousands of variables measured in different social science studies, and it is essential to describe them thoroughly so that the datasets can be understood by other scientists that browse the repository. These variables are, in many cases, the column headers of the data tables present in the dataset.

For easy automated processing, the DDI is defined as an XML Document Type Definition (DTD), and the DDI document uses a DC header in its description.

MODS

The Metadata Object Description Schema (MODS) is a metadata schema for descriptive metadata, derived from MARC 21. It is especially adequate for rich description of electronic resources. Its elements are richer than those of DC, and different from MARC 21 in the sense that their identifiers are not numerical but rather text-based. MODS allows the metadata record to express granular descriptions of the elements that make up the digital object described by the metadata record. MODS is expressed in the XML language [153].

METS

The Metadata Encoding and Transmission Standard (METS) is a metadata standard designed by the Library of Congress (LOC) to represent the structure of objects in a digital library, as well as their administrative and descriptive metadata [153]. It can express the complex relationships between the aforementioned types of metadata, and to do so, it allows the reuse of metadata descriptors present in additional schemas within the several sections of the record [135]. For example, a METS record can contain instances of DC, MARC or PREservation Metadata: Implementation Strategies (PREMIS) elements [219], for example.

METS is formalized as an XML schema, which enables automated validation of the structure of a metadata record, according to the METS specification. Extensive development has been carried out over this format, with many software tools being made available under open-source licenses at the LOC website[8].

LOM

The Learning Object Metadata (LOM) is a metadata standard designed for the description of resources designed to support computer-based learning. The LOM standard divides its attributes into eight categories [153]:

- General, containing information about the object as a whole;

8 http://www.loc.gov/standards/mets/mets-tools.html

- Lifecycle, containing information about the different stages the object goes through during its lifetime;
- Technical, which contains descriptions of the technical requirements and characteristics of the object;
- Educational, which relates to the pedagogical or educational attributes of the object;
- Rights, which describes the IPR and conditions for using the object;
- Relation, identifying objects related to the one being described in the record;
- Annotation, containing comments and the date and author of those comments;
- Classification, which identifies other classification systems for the object.

Accompanying the move from schemas to semantic standards, the principles for translating this schema into RDF have been laid out [156]. A comparison of the advantages and disadvantages of adopting either of the two representations of Institute of Electrical and Electronics Engineers (IEEE) LOM has also been presented: while compatibility with legacy systems is the main reason for adopting a schema-represented version, the advantages provided by Linked Open Data (LOD) are on the favor of a semantic metadata version [3]. Translating the schema representation into a semantic metadata is a hard task, however, but recent works are being carried out in order to profile a complete mapping of LOM into an RDF representation [182]. We will discuss the application of LOD in data repositories in page 31.

2.2.3 Metadata schema crosswalking

The fast growth in digital resources available on the web has motivated widespread developments in metadata schema design, which in turn posed interoperability problems between schemas, records and, as a consequence, repositories. To deal with this issue, the so-called *metadata registries* were created. These databases group sets of metadata descriptors according to the domains in which they are used or their applicability within projects (mandatory, optional, searchable, for example), and try to establish relationships between them [52].

Crosswalking is the process of establishing, when possible, relationships and equivalences between elements from different metadata schemas. These relationships can be one-to-one or many-to-one, in those cases when the source schema is more complex than the target counterpart. This is prevalent in data repositories, which often need to crosswalk their metadata records in MARC XML or MODS to DC, so that they can be harvested through a protocol such as Open Access Initiative—Protocol for Metadata Harvesting (OAI-PMH) [224]. The result of the crosswalking process, called *crosswalk*, is a table that contains those relationships.

Crosswalking enables *metasearching*, which consists in complying with a common protocol (for example, Z39.50), to provide search capabilities records present in systems with different technologies and models [224]. It can also be a good tool to study the actual usage of the descriptors of the different metadata schemas.

During the development of Dendro, our RDM platform, we had to perform crosswalking operations, since we support domain-specific descriptors which are, more often than not, supported in other repositories. When depositing a dataset from Dendro into an external repository, the metadata record present in the Dendro folder had to be converted into a simpler format, for example DC. This was prevalent in almost all the target repositories that Dendro currently supports, whose Application Programming Interfaces (APIs) had little to no flexibility to handle domain-specific metadata. A notable exception was the Comprehensive Knowledge Archive Network (CKAN) platform that, since it supports key-value metadata records in addition to the standard DC fields, allowed Dendro metadata records to be exported as they were recorded in the platform, as sets of Descriptor URI - value pairs.

Ontologies can help greatly with the implementation of the crosswalking process, since the relationships between descriptors can be expressed easily in Web Ontology Language (OWL). At the same time, reasoner engines can take into account the relationships between descriptors, and then use inference to automatically perform the crosswalking operation when a query is performed over several knowledge bases that use different descriptors to represent related concepts in their records.

2.2.4 Application Profiles

Since there is a multitude of research domains in existence, the needs of their user groups also differ regarding their needs for metadata. As a consequence, it is often hard to retrieve metadata schemas that suit their needs. This is sometimes impossible, since existing schemas are in many cases suited to satisfy the needs of a broad community instead of the needs of individual groups [52]. As a response to this, the concept of Application Profile (AP) emerged as a way to satisfy individual needs of highly specialised user groups [69, 118]. APs are sets of "data elements drawn from one or more namespaces combined together by implementors and optimized for a specific local application" [94]. This *mix-and-match* approach resembles the best practices of ontology design, where concepts should first be reused from existing ontologies and combined or derived to design a new ontology for the domain being modeled [35]. This approach has been followed in recent studies [47], which have shown that it can be hard for researchers to manually outline such metadata recommendations to describe their datasets. A possible solution could be to devise a system capable of creating application profiles in a semi-automated way [194], while the curator would validate metadata descriptions at the time of deposit in an institutional repository. This "self-service" approach to data curation has already been adopted in the past [204, 101].

The DCMI defines an Application Profile as "a declaration of which metadata terms an organisation, information resource, application, or user community uses in its metadata". An AP must comply with the following requirements [19]:

- It cannot declare new metadata terms and definitions, instead only reusing terms from existing element sets is permitted.
- The included elements must be uniquely identified by URIs within XML namespaces.

- Any new term coined for use in an AP must first be declared in a form that can be cited by the AP—in other words, without a citable for of a metadata schema element, it is impossible to properly reuse it in the AP. This can be obtained through persistent URIs.

- The AP may provide documentation on on how the terms selected are constrained, encoded or interpreted in the context of the particular application for which it is designed.

APs can resort to several mechanisms to satisfy the domain-specific needs that they are designed for, without breaking the aforementioned restrictions that dictate that only reuse and refinement is allowed, ruling out the definition of any new terms [69]:

- **Cardinality enforcement**
  By imposing different cardinality restrictions on existing metadata elements, an AP can change the requirements for those elements. For example, a descriptor that is specified as optional in the source metadata schema can be specified as required or conditional in the AP. These specifications must always operate under the constraints of the schema from which the metadata elements originate; it cannot relax restrictions imposed by it, or else it will jeopardise the interoperability of the AP-compliant records with the ones that comply with the original schemas.

- **Value space restriction**
  The AP can impose stricter requirements for the acceptable values of a metadata element. For example, if an element specifies that its valid values are names of individuals, an AP that reuses such an element may specify that the valid values for that element can only be names of people, but with the surname first, followed by a comma and then the initial of the first name. Another example can be an element whose valid values are dates (a character string without further restrictions). An AP that reuses the element can specify that the values must be specified as ISO-8601-compliant date representations.

- **Relationship and dependency specification**
  The AP can enable schema designers to specify stricter requirements regarding the relationships between different elements, either from the same schema or other schemas. For example, the AP designer can specify that if a certain element is included, another must also be included. The possible values of an element may also be restricted according to the value of another. As long as these restrictions do not relax the ones imposed by the the source schemas from where the elements are drawn, the AP will be correct.

- **Declaration of namespaces**
  APs allow the specification of multiple namespaces, enabling researchers to include descriptors from different schemas, separating them through the use of different namespaces. Local elements can also be added to a local namespace to distinguish them from existing ones.

2.3 ONTOLOGY-BASED APPLICATION PROFILES

Schemas can be expressed in XML or RDF, with XML being more suited for specifying the structural and syntactic constraints of metadata records, while RDF supports rich semantic descriptions that XML schemas cannot provide. Attempts to combine the two have been presented, but implementers usually opt for one of the two [110].

It is interesting to see that there is a move towards semantic metadata in current repository platforms. DSpace suports the usage of many metadata schemas in its metadata records, but the records are mainly flat and search capabilities are limited to free text or field-based search. The need for semantic search and richer interoperability has led to the development of semantic models for metadata representation in this widely used repository platform [125].

The development of ontology-based APs still remains an under-explored approach, with XML schemas still assuming a prominent role—perhaps due to the difficulties in finding base ontologies on which to base the APs on.

2.3.1 Applications of Application Profiles as ontologies

According to the classic definition by Heery & Patel, APs are designed by combining descriptors from different metadata schemas in order to satisfy the requirements of a specific application. They can refine existing definitions or restrict permitted values within the domains of the existing elements but they may not define new elements [94]. These constraints are very easily represented using ontologies, which by definition are designed for easy sharing and reuse on the web [30]; new concepts can be defined as subclasses of subproperties of existing ones.

It makes sense, thus, to represent an AP as an ontology, making it a *semantic application profile*. The development of this type of APs can start with the mapping of the existing application profile into a OWL representation, which is then followed by a refinement of the translated model according to the needs of the domain [124], while in other cases dedicated tools have been designed over APs built by ontologies, namely the LOM ontology [28].

We identified the lack of a standard—or at least broadly supported—way to represent an AP as an ontology itself. This would require the definition of a meta-ontology for AP representation, something that we outline as part of our future work.

2.3.2 Dataset granularity in the presented approach

It is entirely different to develop an ontology for representing metadata or for representing the dataset itself as a set of triples. Therefore, we must define the granularity of the objects that we will be describing throughout this work.

Presently, it is hard to find OWL-represented ontologies that can be used as the basis for metadata production at the level that our work requires—to describe the files and folders and not the data itself. This is because many available ontologies are very fine-grained and focused on modelling concepts of a domain. For example, an ontology may model a Measurement class, with several properties having it as their domain. This means that a table contained in a dataset must usually be modeled as a series of Measurements (one per line), and the columns modelled as instances of

properties also defined in the ontology. This is operationally unwieldy and would simply overwhelm the supporting triple store database after a few dataset deposits, since in a research environment it is easy to encounter Excel spreadsheets with hundreds of thousands of lines, for example, that would have to be modeled as a unique resource each.

This granularity is therefore incompatible with our proposed approach, making us exclude many ontologies currently available online. We will discuss this in more detail on page 111. In contrast with this approach, throughout this work we consider a dataset to be a data file, or a folder which contains a series of data files.

In order to represent the metadata related to the dataset, we should not need to model its contents, but instead consider it as a unique, identifiable resource, without decomposing it. If, however, we were to consider mapping the contents of the files into triples or datasets and sub-sections of datasets as new resources, we could consider alternatives such as the Vocabulary for Interlinked Datasets (VoID) ontology [7].

2.4 ONTOLOGY REPOSITORIES

There are many ontologies available on the web, and unfortunately scattered across it. To cope with this issue, special purpose platforms have been created to facilitate their discovery and retrieval. While some ontology repositories follow a relatively fast and simple depositing procedure, others require more detailed metadata records for the deposited ontologies and the deposit process includes manual validation stages.

2.4.1 Linked Open Vocabularies

Linked Open Vocabularies (LOV) is a portal designed for sharing semantic vocabularies on the web. It focuses on high-quality ontology documentation and sharing via machine-friendly formats such as LOD or SPARQL Protocol and RDF Query Language (SPARQL) as well as human-friendly methods (easy to use User Interface (UI) combined with full-text search). It is a "high-quality catalogue of reusable vocabularies for the description of data on the web", helping its users "determine which vocabularies to use to describe the semantics of their data" [218].

LOV includes analytics software that can provide valuable information on the characteristics of the vocabularies within the catalogue (classes, predicates or individuals). Their usage patterns are also studied, not only in terms of accesses (by humans and machines) but also in terms of concept and vocabulary reuse—the portal can track, for example, which concepts (and therefore, which vocabularies) are reused in other vocabularies. How a vocabulary is reused is an important insight, with possible usages contexts being the production of *Metadata* (a concept from a vocabulary is used in the production of metadata for another vocabulary), *Import* (some terms of a vocabulary are reused to capture the semantics of other terms), *Specialization*, *Generalization*, *Extension*, *Equivalence* and *Disjunction*.

Curation of the vocabularies is carried out in a semi-automated fashion from the moment of deposit. Striking a balance between volume and quality, the workflow relies on an automated deposit, followed by a manual review to ensure that the deposited vocabularies match the quality requirements placed by the LOV catalog:

- They should be written in RDF.
- Should be parseable without errors.
- All terms should have an rdf:label.
- Should reuse relevant vocabularies.
- Basic metadata should be present (at least a title).

LOV has a central role in the lifecycle of vocabularies, as highlighted by the World Wide Web Consortium (W3C)[9].

BioPortal

BioPortal is the largest repository for ontologies used in the Biomedical science domain [198]. It provides interesting features such as metrics, peer review and classification of the available ontologies according to their specific domains. The available metrics include the number of classes, predicates and individuals, among others. Peer review features include comments on different aspects of the deposited ontologies: degree of formality, documentation and support and usability are some of the aspects on which the peers can review the deposited ontologies. The domains and categories allow for easy discovery of ontologies upon deposit, helping reduce redundant ontologies or concepts in the repository.

Ontologies can also be made public, private or licensed. The visibility is controlled by the owners of the ontologies, and enforced through an API key that any clients trying to access the ontologies must provide in their Hyper Text Transfer Protocol (HTTP) requests.

Cupboard

Cupboard is an ontology repository designed to help users deposit, comprehensively describe and find ontologies. Cupboard focuses in providing advanced search mechanisms which allow users to find and explore ontologies, using a visual browser to picture the relationships between the concepts in the ontology. This work defines an *ontology space* as a separate ontology index, which uses the Watson[10] search engine [64] as its implementation. Ontology spaces are useful for exposing only certain sets of ontologies for building applications, for example [62, 63].

The platform also implemented a module for establishing mappings between concepts of different ontologies, or *ontology alignments*, which can be used to populate ontology spaces in addition to the ontologies themselves.

Integration with external systems has been a concern in the development of Cupboard, with the inclusion of a comprehensive API. Programmatic interfaces have been implemented for the main operations, such as: creating an ontology, adding metadata to an ontology, managing ontology alignments, among others. Cupboard implemented a plugin to integrate with NeON Toolkit[11], an ontology development environment. User experiments have shown that this integrated ontology development solution lowers the cost of building an ontology, while improving the quality of the finished results.

9 http://www.w3.org/2013/data/
10 http://watson.kmi.open.ac.uk/WatsonWUI/
11 http://neon-toolkit.org

Figure 3: Prefix.cc search interface and results

Prefix.cc

A very common problem in ontology design, as well as in the adoption of semantic metadata records in operational systems, is to determine if a given prefix has already been adopted for a given ontology, and if that is the case, which ontology does it stand for.

Prefix.cc[12] provides a web-based service which allows users to search for a prefix, for example foaf and obtain a list of URIs for ontologies that have been associated to that prefix. Every element in the list can be voted up or down by the users to help disambiguate between the different alternatives. This community-based approach at disambiguation makes it possible to separate the most used ontologies from those that—although adopting the same prefix in some records—-are not as widely recognized by that prefix.

Figure 4 shows two screenshots from the prefix.cc portal, showing the list of results of two queries, one for the foaf prefix and another for the geo prefix. The top image shows the result of voting in one of the results in the list.

EUDAT

The integrated data management environment EUDAT provides a module (B2Share) for depositing and sharing ontologies in an easy way. In fact, the ontologies developed throughout the course of this work have been deposited in EUDAT for public consultation and reuse[13].

12 See http://prefix.cc
13 https://b2share.eudat.eu/record/292

2.5 CONCLUSIONS

Metadata is an essential part of the description and discovery of resources. From informal records to semantic metadata represented as RDF, the different degrees of formalism of metadata records, as well as their supporting technologies have evolved greatly.

Semantic metadata supported by ontologies provides, beyond structural correctness, a precise account on the meaning of the descriptors. Ontologies allow not only the easy sharing of interoperable records built on standard representation formats, but also of the ontologies which carry the meaning of the descriptors themselves. These are hard to manage by non-experts, but can be shared easily on the web, and able to be automatically processed by machines. Our RDM platform, Dendro, is built on ontologies, allowing it to represent metadata in a very flexible way. Through the platform, LOD-represented metadata records, as well as ontologies can be shared to prominent repositories, along with datasets.

An issue with this approach is that it can be hard to retrieve the ontologies that contain relevant descriptors for data description, because they are not easily discoverable. In particular, the lightweight ontologies which do not try to model the contents of data files but instead refer to the files themselves, which are the atomic resources to be described in this work. Ontology repositories are therefore an essential part of the Dendro ecosystem. Dendro administrators need to use these platforms as sources of ontologies for the platform, while Dendro could facilitate ontology sharing by representing the APs it generates as ontologies, so that they can be deposited in an ontology repository along with any necessary metadata.

The transition from records supported by XML-based metadata schemas to semantic metadata LOD represented as RDF documents is underway. As we will explore in the next chapter, RDM platform designers have realised that DC is sometimes not enough to comprehensively describe research datasets, calling for the implementation of APs and domain-specific metadata schemas for dataset metadata records. This is where LOD and ontologies can assist by fostering reuse and interoperability.

3 PLATFORMS FOR RESEARCH DATA MANAGEMENT

In this chapter, we present an overview of several prominent research data management platforms that can put in place by an institution to support part of its research data management workflow. We start by identifying a set of well known repositories that are currently being used for either publications or data management, discussing their usage in several research institutions.

We focus on the needs of the different stakeholders involved in a data management workflow. Through the requirements of each stakeholder, we analyse their fitness to handle research data, including variable metadata requirements and compliance with preservation guidelines. We also take into consideration their implementation costs, architecture, interoperability, content dissemination capabilities, implemented search features and community acceptance.

3.1 INTRODUCTION

When faced with the many alternatives currently available, it can be difficult for institutions to choose a suitable platform to meet their specific requirements. Several comparative studies between existing solutions were already carried out, in order to evaluate different aspects of each implementation [76, 15, 23]. This evaluation considers aspects relevant to this work, focused on finding solutions to research data management, and taking into consideration our past experience in this field [196]. This expertise allows us to have insights on specific, local needs that can greatly influence a platform's adoption and thus, its implementation success.

It has been shown that the absence of timely description from the start of data production can yield lackluster descriptions [145]—a very practical example is when researchers leave their teams after publishing, but without describing their datasets. A possible solution would be to have data curators accompany the research workflow—however, small research groups may struggle to keep up with the description demands posed by the existing datasets, and we may not expect researchers to spend much time in data description activities. A possible compromise scenario is to support researchers in the description of their data as they produce them, postponing curator intervention until later in the workflow.

There are several factors contributing to data loss:

1. Lack of resources for data curation

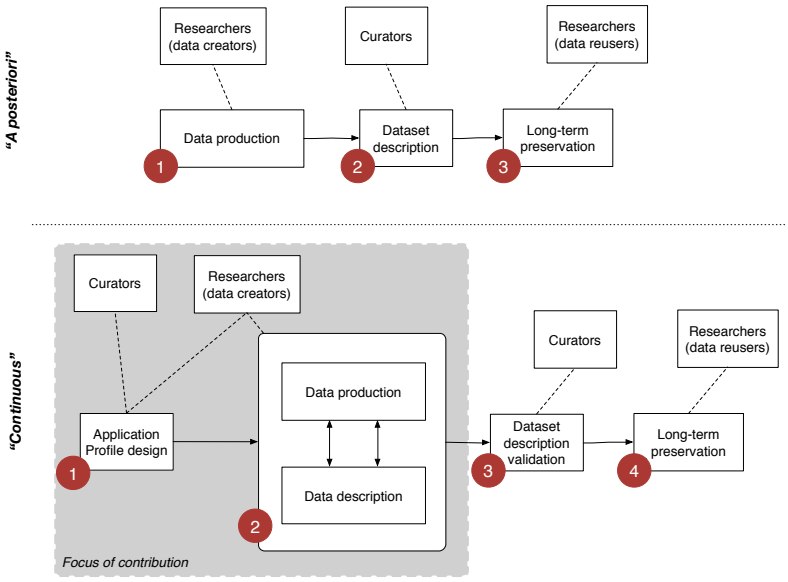


Figure 4: Scope of the contributions of this work in the context of *a posteriori* data description

2. Overwhelming numbers of datasets for small research groups to manage by themselves
3. Diversity of domains, requiring domain-specific metadata schemas or Application Profile (APs)
4. High turnover of researchers within research teams, worsening data loss.

In the long tail of research, the management of research data products is often carried out by elements of the research team itself. Storage solutions are often limited to external hard drives or network drives, and there are no established means to produce metadata records. When they exist, these informal metadata records often consist of "README" files that are bundled with the data files. The contents of these files is written by the researchers themselves, without a standard representation for the metadata, which limits the possibilities of automated ingestion, indexing and retrieval by potential re-users [211].

Figure 4 highlights the differences between the traditional *a posteriori* data management workflows and what we named *collaborative* data description. While in the first approach the researchers work completely separately from curators, the second approach requires an initial cooperation effort between the two parties. The reasons for this are the need to provide researchers with basic knowledge on how to manage their data, as well as to build or select (if they exist) the metadata schemas that will be adequate to the description of datasets from each researcher's particular domain. After this setup phase, researchers themselves will be able to perform data description as they produce their datasets. On the *a posteriori* approach, conversely, data description is relegated to a later moment and is usually performed by the curator—in the *collaborative* workflow, the curator has the task of validating the metadata upon deposit, instead of having to produce it from scratch. In order to anticipate the start of the data description works, it is necessary to reduce the dependency on the curator throughout the entire lifecycle of

the data—and such goal can only be attainable if their work can be partially automated.

This chapter is based on the following publication [13]:

- Amorim, R., Castro, J., Rocha, J., & Ribeiro, C. (2015). A Comparative Study of Platforms for Research Data Management: Interoperability, Metadata Capabilities. New Contributions in Information Systems and Technologies, 1, 101–111. http://doi.org/10.1007/978-3-319-16486-1

3.2 CAPABILITIES OF EXISTING RESEARCH DATA MANAGEMENT SYSTEMS

Managing research data is gradually becoming a common practice to ensure that the research outputs are recorded and preserved over time [141, 188]. Currently, several research teams resort to known platforms to manage their data and ultimately share it with the research community, while also guaranteeing their preservation.

In order to be considered as a possible candidate for inclusion in our proposed data management workflow, a data management platform should satisfy several requirements. Since we target our workflow at small research groups with limited funding and resources for data curation, the present comparison may be applicable to many research groups beyond those included in our panel.

While some of these platforms implement features specifically targeted at the description of datasets, others follow approaches closer to the established description practice for research papers. There are several system design recommendations currently in place that can help us better understand how a data management system should be implemented. The best practices also extend to metadata record representation and interoperability. This chapter provides an overview on these aspects.

3.2.1 Open-source versus proprietary solutions

There are advantages of an open-source solution when compared to a proprietary alternative, namely on the possibilities of future maintenance, developments and updates. Being open source does not guarantee the indefinite survival of the software solution *per se*; however, it enables third-party developers to contribute to the code base, keeping it updated in order to deal with security problems, bugs and performance issues. Compatibility with emerging metadata standards and existing systems can also be more easily achievable—as an example, if an institution wishes to integrate a data management solution with an existing authentication system or incorporate datasets into its publications management workflow, it can give the task to its own developers and IT team, instead of paying for consulting from a service provider. Moreover, those code contributions that are initially built to cope with the requirements of a particular institution, but then considered interesting to other institutions can easily be incorporated into the main codebase of the platform and become available as updates to everyone using the open-source platform.

As a result of these increased chances for development, an open-source platform may outlive its proprietary counterparts. Also, since its code is public, the recovery of the data stored within or the migration to another

platform can be carried out more easily, as it is possible to determine exactly how it uses the information stored in the underlying data storage layer by looking at the code for its business logic. The open-source characteristic is not, however, a necessary condition for successful decommissioning of a data management software platform. While some may argue that the problem of decommissioning can be handled by adhering to the Open Archival Information System (OAIS) guidelines and the implementation of Dissemination Information Packages (DIPs), the fact remains that there is no single, standardised format for DIPs. As a result, the DIPs built by a software platform may not be compatible with any other platforms, requiring the implementation of translation logic to carry out the conversion steps for the migration.

Another important reason for the adoption of an open-source platform in detriment of a proprietary one is to avoid vendor *lock-in*. Vendor *lock-in* occurs when an institution adopts a proprietary solution and, from then on, is dependent on the company that built the system for support and maintenance. The problem is exacerbated if the proprietary solution does not provide a standards-based way of exporting or migrating the data and metadata stored within to a new solution. In a data management environment supported by software solutions, a crucial aspect is to ensure that the data survives the obsolescence of the software in which it is stored. Vendor *lock-in* hampers this, and places the data at risk in the event of the vendor company ceasing to exist, since support would be discontinued. Without access to the platform's source code, it would be impossible for anyone to take over the development efforts and prolong the platform's operational life or, at least, develop ways to facilitate the migration of data and metadata stored in the instances of that platform into a more modern solution (if such migration mechanisms are not already in place by that time).

3.2.2 Full control over the data

Institutions and researchers are often unwilling to deposit their data in places outside their control, such as commercial cloud storage platforms. The main reasons for this include the manipulation of sensitive data (clinical data, for example) or the potential loss of Intellectual Property Rights (IPR) or of the innovative nature of the research efforts in question.

The problem could be reduced by the application of encryption to all the data stored outside of the organization, with automatic encryption of outgoing data and decryption of incoming data. This solution places an additional layer of complexity over the solution, opens the possibility of losing access to the whole data altogether (in the case of losing a decrypting key, for example), and is computationally expensive for files that are continuously manipulated.

An alternative is to allow researchers and institutions to host their own instances of a data management environment behind an institutional firewall, and therefore invisible to the outside world. This way, the information would be always accessible to those individuals within the institutional network and no data would be stored outside of institutional control.

3.2.3 OAIS Model

The OAIS reference model is a model for the creation of an OAIS [59]. The model introduces the notion of Representation Information of a data re-

source, defining it as "the information that maps a Data Object into more meaningful concepts". Another relevant notion is *long-term preservation*, which aims to ensure the accessibility to the preserved data even past the time of obsolescence of the OAIS itself. Overcoming changing technologies, including media types and formats, is an important concern to indefinitely maintain access to the preserved data.

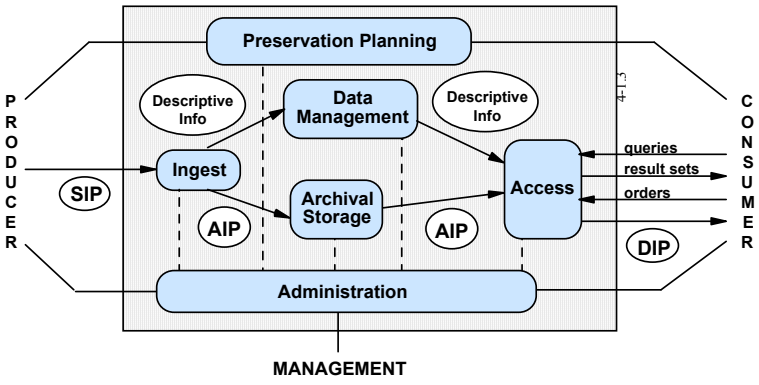


Figure 5: "OAIS Functional Entities" (image retrieved from [59])

Figure 5 shows an overview of the flow of data inside a OAIS and will now be briefly described. There are three main formats used to represent data: Submission Information Package (SIP), Archival Information Package (AIP) and DIP.

Ingestion

In order to comply with the OAIS reference model, a system must support the ingestion of data in the form of SIPs. This means being able to ingest the data in that format while preparing it for storage and management within the archive. Quality assurance steps must be taken throughout the process of transforming these SIPs into AIPs that may be passed on to the Archival Storage entity for deposit so that they can be stored and later managed by the Data Management logic.

Archival Storage and Data Management

OAISs-compliant systems must provide a set of services and functions capable of receiving AIPs from the ingestion endpoint, recording the data that they contain while checking and correcting errors. Disaster recovery capabilities are also mandatory. All these features must be accessible via Application Programming Interfaces (APIs).

Access

The Access layer must provide a set of endpoints for the discovery, description and retrieval of the information stored in the OAIS. This allows other systems to consume the data stored in the system, which must be provide in the form of DIPs.

Supporting activities

An OAIS system must include management and planning aspects to support the described workflows and mechanisms. These are divided into two components: *Preservation Planning* and *Administration*.

Preservation Planning comprises all tasks that are required to ensure the long-term preservation of data, even past the operational lifetime of the operational system. This activity includes continuously *monitoring* the status of the data, ensuring timely *migration*, which is necessary to maintain data accessible. For example, file formats can become outdated. A possible solution to these issues is to accompany their evolution by migrating the data stored in an old format to the new one, keeping the original and the code that performed the migration, as well as other details on the process. *Continuous improvement* of data management policies completes the set of tasks that make up the Preservation Planning activity. In this regard, and given its results [120], we can consider the Data Asset Framework (DAF) guidelines [121] as a good place to start when implementing an OAIS-compliant data management policy.

Administration is concerned with all the services required to maintain the system in effective working order, helping producers deposit data, auditing the deposit process to ensure high data quality standards, as well as carrying out hardware and software migrations whenever required. It is also responsible for ensuring that the policies selected by the Planning module are in fact followed and providing support to all who interact with the system.

The 2009 report by Beagrie et al. [26], which described the cost/benefit ratio implied in annotating resources with metadata, derives notions from the OAIS reference model. On a more technical note, prominent data archival and preservation platforms such as the UK Data Archive use the workflow specified by the OAIS reference model [27].

3.2.4 OAI-PMH protocol

The Open Access Initiative—Protocol for Metadata Harvesting (OAI-PMH) [128] was designed to facilitate the harvesting of metadata records by entities that are external to the system where the records are stored. Perhaps as a result of its presence in many open-source repository software platforms, it achieved a wide user base, with almost 1800 institutional repositories worldwide implementing the protocol, as of 2008 [93].

While the protocol provides an interoperable way to harvest metadata records from different repositories, it also carries with it a specific set ot technical requirements. The need to implement a specific API for OAI-PMH compliance (for example, being able to respond to the specifying OAI-PMH *Requests* such as *Identify*, *GetRecords*, among others). There are six mandatory Requests that an enpoint must implement in order to be OAI-PMH compliant [67], and each request must be met with a specific type of *Response* (see Figure 6).

OAI-PMH is implemented in many open-source repositories such as DSpace, Fedora and ePrints; when compared to Linked Open Data (LOD), however, it appears complex and unwieldy. LOD does not rely on specific *Requests* and *Responses*, instead opting for using already existing mechanisms such as *content negotiation* to adequately *de-referentiate* resources, dispensing the need to implement a protocol that is arguably more specific and complex.

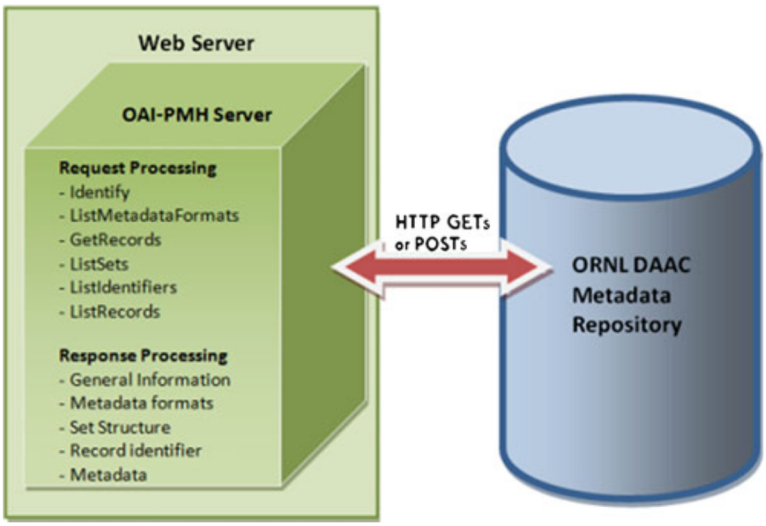


Figure 6: "OAI-PMH Overview" (image retrieved from [67])

When comparing the querying capabilities of *OAI-PMH* versus those of a *LOD*-enabled solution (complete with a SPARQL Protocol and RDF Query Language (SPARQL) endpoint), it is clear that *SPARQL* endpoints provide a much higher degree of flexibility and capability.

SPARQL provides a full querying language where restrictions can be specified over both values and the types of *terms* that make up the metadata records (e.g. retrieving items that only have "João" in their Dublin Core (DC) contributor instances); OAI-PMH, on the other hand, requires that all the records be retrieved, stored and indexed in a queryable form—it is the job of the client to make the records queryable, since the protocol is only targeted at providing records to the outside world in an standard form, not necessarily to make them queryable on the server side. These shortcomings of OAI-PMH when compared to LOD have been highlighted in the past: the protocol does not make its resources available via dereferencable Uniform Resource Identifiers (URIs) and provides very limited ways for selective access to metadata (in other words, metadata record querying capabilities are limited) [93].

3.2.5 Linked Open Data

Integrating different sources of data is currently a major trend in the field of digital preservation. The benefits are clear for all stakeholders in the research environment. For the institutions that support the repositories, the ability to showcase their research production to their target audience (typically researchers looking for datasets and related publications) in a transparent way can help them gain recognition through peer citation. For authors, data sharing can foster collaboration between different research groups while improving research reproducibility and making it easier to compare results [36].

Easier retrieval of relevant data from different sources in a way that is transparent to the end-user is one of the promises of LOD, that aims at helping machines link resources on the Web by establishing the *semantics* of those links [31]. Widespread LOD compliance would make it possible

to interlink the resources contained in research data repositories, improving current metadata-based retrieval approaches.

The major open-source repository platforms are expressing interest in exposing their metadata records through LOD-compatible endpoints. Currently, they all implement OAI-PMH endpoints for metadata exposure. This protocol allows other systems to interact with the repository to gather metadata records but poses several issues that limit the interoperability of these approaches.

Until LOD-compliant metadata exposure techniques and transparent federated search functionality become current in research repositories, transient solutions for metadata harvesting and record interlinking have been developed. Some of these approaches include Mercury, a metadata harvesting suite used by NASA [68] and the OAI2LOD [93, 92] server, a *middleware* solution designed to mediate the communication between LOD clients and OAI-PMH endpoints.

3.2.6 Persistent URIs

Persistent Identifiers are associations between a character string and a resource. The persistent identifier plays a very important role in the research data management workflow by enabling that resource to be referred by another resources. Repositories *mint* these identifiers to allow their users to refer (or *cite*) the resources that they host.

The identifiers also allow a repository to change its internal storage and retrieval mechanisms that allow resources to remain available, while ensuring that the citations that point to that resource always remain valid. The repository can therefore migrate its software platform when it becomes obsolete, with minimum consequences [61]. The DataCite consortium[1] refers persistent identifiers as a "key to their service", which aims to provide a *persistent* approach to access, identification, sharing and re-use of datasets [53].

While the identifier's role is to provide a persistent link to a resource, it is up to the repository to ensure that the resource remains available, unchanged, and that its *provenance* can be determined. The identifier assumes that these requirements are present.

Perhaps the most widely known identifier system for digital objects is the Digital Object Identifier (DOI) system [162]. It is a separate system from the data management solutions themselves, but whose API is tasked with minting URIs which point to the resource in question. All major publications now mint DOIs as the main digital identifier to allow reliable citation.

3.3 COMPARING RESEARCH DATA MANAGEMENT PLATFORMS

For dataset management purposes, there are several additional solutions. From these, we highlight the Comprehensive Knowledge Archive Network (CKAN)[2], Zenodo[3] Figshare[4] and Dryad[5].

1 http://www.datacite.org
2 http://ckan.org
3 http://zenodo.org
4 http://figshare.com/
5 http://datadryad.com/

This comparison covers open-source solutions that can be *both* deployed on the cloud or in a research institution's servers. There are many platforms currently available, so we first need to adequately set the scope of our analysis.

The typical scenario of application of our application profile generation approach is a small research institution or group that does not possess the financial resources to support a curatorial team, and can only have limited IT staff support. In these circumstances, it is not uncommon to see a member of the research team itself take the role of data curator, supporting the research team however possible. This sort of scenarios can be classified as part of the long-tail of science.

In the long-tail of science there are many research groups that produce huge numbers of small datasets. Given their limited resources and data management expertise, they can neither hire curation services, and are also unable to manually build application profiles to satisfy their data management requirements while ensuring a high degree of interoperability and the comprehensiveness of the produced metadata records.

3.3.1 From publications to data management

The growth in the number of research publications combined with a strong drive towards open access policies [43, 54] continues to drive the development of several open-source platforms for managing bibliographic records. While data citation is not yet a widespread practice, the importance of citable datasets is growing. Until a culture of data citation is widely adopted, however, research groups are opting to publish so-called *data papers*, which are more easily citable than datasets. Data papers serve not only as a reference to datasets but also document their production context [45].

There are already several good alternatives for publications management, as documented in a recent comparison of sofware designed for Institutional Repositories published by UNESCO, which covered Digital Commons[6], DSpace[7], ePrints[8], Fedora[9] and Islandora[10] [23]. These are primarily designed for depositing, indexing and retrieving academic publications and cultural artifacts. From these, Islandora is directly related to Fedora, as the former is an extended version of the latter, complete with extensions based on other open-source software such as the Drupal Content Management System[11].

As data management becomes an increasingly important part of the research workflow [141], solutions designed specifically for research data are being actively developed by both open-source communities and data management-related companies. As with publication repositories, many of their design and development challenges aim at the description and long-term preservation of research data. There are, however, at least two fundamental differences between publications and datasets: the latter are often purely numeric, making it very hard to derive any type of metadata by only looking at their contents; also, datasets require detailed, domain-specific descriptions in order to be interpreted. Metadata requirements can also vary greatly from domain to domain, requiring repository data models to be flexible

6 http://digitalcommons.bepress.com/
7 http://www.dspace.org/
8 http://www.eprints.org/uk/
9 http://fedorarepository.org/
10 http://islandora.ca/
11 https://www.drupal.org/

enough to adequately represent these records [197]. The effort invested in adequate dataset description is worthwhile, since it has been shown that research publications that provide access to their base data consistently yield higher citation rates than those that do not [170, 171].

As these repositories deal with a reasonably small set of managed formats for deposit, several reference models, such as the OAIS [59] are currently in use to ensure their preservation and promote metadata interchangeability and dissemination. Besides capturing the available metadata during the ingestion process, they often distribute this information to other repository instances, improving visibility of the publications through specialized research search engines or repository indexers. While the former focus on querying each repository for exposed contents, the latter help users find data repositories that match their needs—such as finding repositories from a specific domain or storing data from a specific community. Governmental institutions are also promoting the disclosure of Open Data to improve citizen commitment and government transparency, and this motivates the use of data management platforms in this context.

3.3.2 Methodology

For this comparison, we have selected data management platforms with instances running at both research and government institutions, namely DSpace, CKAN, Zenodo, Figshare, ePrints, Fedora and EUDAT. If the long-term preservation of research assets is an important requirement of the stakeholders in question, other alternatives such as RODA [183] and Archivematica may be considered strong candidates, since they implement comprehensive preservation guidelines not only for the digital objects themselves but also for their whole life cycle and associated processes. On one hand, these platforms have a strong concern with long-term preservation by strictly following existing standards such as OAIS, PREMIS or METS, which cover the different stages of a long-term preservation workflow. On the other hand, such solutions are usually harder to install and maintain by institutions in the so-called *long-tail of science*—institutions that create large numbers of small datasets, but that do not possess the necessary financial resources and preservation expertise to support a complete preservation workflow [95].

We also highlight the Fedora framework[12], which is used by some institutions, and is also under active development, with the recent release of Fedora 4. The fact that it is designed as a framework to be fully customized and instantiated, instead of being a "turn-key" solution, places Fedora in a different level, that can not be directly compared with the existing solutions. Two open-source examples of Fedora's implementations are Hydra[13] and Islandora[14]. Both are open-source, capable of handling research workflows and centered around the best-practices approach already implemented in the core Fedora framework.

An overview of the previously identified stakeholders led us to select two important dimensions for the assessment of their capabilities: their architecture and their metadata and dissemination capabilities. The former includes aspects such as how they are deployed into a production environment, the locations where they keep their data, whether their source code is available

12 http://www.fedora-commons.org/
13 http://projecthydra.org/
14 http://islandora.ca/

Table 1: Limitations of the identified repository solutions

	Instance count▽	Closed source	No API	No unique identifiers	Complex installation or setup	No OAI-PMH compliance
CKAN	116*			✗†		✗†
ContentDM	53	✗				
Dataverse	2					
Digital Commons	141	✗	✗			
DSpace	1272					
ePrints	397			✗†		
EUDAT	—	✗‡				
Fedora	41				✗	
Figshare	—	✗				
Greenstone	51		✗	✗	✗	
Invenio	No data					
Omeka	4			✗		✗†
SciELO	18	✗				
WEKO	38			No data		
Zenodo	—					

▽Extracted from the OpenDOAR platform, as of July 2015. *According to the corresponding website. †Only available through additional plug-ins. ‡ Only partially.

or not, and other aspects that related to the compliance with preservation best practices. The latter focuses on how resource-related metadata is handled and the level of compliance of these records with established standards and exchange protocols. Other important aspects are their adoption within the research communities and their available support for possible extensions.

3.3.3 Existing repositories

Managing research assets is a complex task. While the process for deposit and access to publications from different domains is well established in most institutions, ensuring the same level of accessibility to data resources is still a concern, and different solutions are being experimented to expose and share data in some communities. Table 1 lists some well-known platforms that are currently being used by institutions worldwide. This preliminary analysis identifies features of the platforms that may render them inconvenient for data management. To build the table, we resorted to the documentation of the platforms, and to basic experiments with demonstration instances, whenever they were available. In the first column, under "Registered repositories", we have the number of running instances of each platform, according to the OpenDOAR platform as of mid-July 2015.

In the analysis, we consider five evaluation criteria that can be relevant for an institution to make a coarse-grained assessment of the solutions. We excluded some existing tools from this first analysis, mainly because some

of their characteristics place them outside of its scope. This is the case of platforms specifically targeting research publications (and that cannot be easily modified for managing data), heavy-weight platforms targeted at long-term preservation. We also exclude those that, from a technical point of view, do not comply with desirable requirements for this domain such as following a open-source approach, or providing access to their features via comprehensive APIs.

By comparing the number of existing installations, it is natural to assume that a large number of instances from a platforms is a good indication of the existence of support for its implementation. Repositories such as DSpace are widely used among institutions to manage publications. Therefore, an institution using a DSpace installation to manage publications can profit of its expertise with this platform to expand or replicate the repository and meet additional requirements. It is important to mention that some repositories do not implement interfaces with existing repository indexers, and this may cause the OpenDOAR statistics to register a value lower than the actual number of existing installations. Moreover, services provided by EUDAT, Figshare and Zenodo, for instance, consist of a single installation that receives all the deposited data, rather than a distributed array of manageable installations.

Government-supported platforms such as CKAN are currently being used as part of the Open Data initiatives in several countries, allowing the disclosure of data related to sensitive issues such as budget execution, and their aim is to vouch for transparency and credibility towards to tax payers [127, 107]. Although not specifically tailored to meet research data management requirements, these data-focused repositories also have an increasing number of instances supporting complex research data management workflows [223], even at universities[15].

Access to the source code can also be a valuable criterion for selecting a platform, primarily to avoid vendor lock-in, which is usually associated with commercial software or other provided services. Vendor lock-in is undesirable from a preservation point of view as it places the maintenance of the platform (and consequently the data stored inside) in the hands of a single vendor, that may not be able to provide support indefinitely. The availability of the a platform's source code also allows additional modifications to be carried out in order to create customized workflows—examples include improved metadata capabilities and data browsing functionality.

Commercial solutions such as ContentDM may also incur high costs for the subscription fees, which can make them cost-prohibitive for non-profit organizations or small research institutions. In some cases such as EUDAT, only a small portion of the source code for the entire solution is actually available to the public. In this particular case, only the B2Share module (targeted at the deposit and dissemination of datasets in the long-tail of science) is currently open[16]—the remaining modules are currently unavailable.

From an integration point of view, an existing API can prove to be a source of continuous developments and help with the repository's maintenance, as the software ages. Solutions that do not meet, or partially comply with this requirement, may hinder their integration with platforms that can improve the existing contents' visibility. The lack of an API creates a barrier to the development of tools to support a platform in specific environments, such

15 http://ckan.org/2013/11/28/ckan4rdm-st-andrews/

16 Source code repository for B2Share is hosted via GitHub at https://github.com/EUDAT-B2SHARE/b2share

as a laboratories that frequently produce data to be directly deposited and disclosed. Finally, regarding long-term preservation, some platforms fail to provide unique identifiers for the resources upon deposit, which makes them hard to persistently reference or cite in publications, for instance.

Support for flexible research workflows makes some repository solutions attractive to smaller institutions looking for solutions to implement their data management workflows. Both DSpace and ePrints, for instance, are quite common among research institutions to manage their own publications, as they offer broad compatibility with exchange protocols (OAI-PMH) and preservation guidelines such as the OAIS model, which requires, among other things, the existence of different SIP, AIP and DIP packages.

3.3.4 Stakeholders in research data management

Several stakeholders are involved in dataset description throughout the data management workflow, playing an important part in their management and dissemination [141, 36]. These stakeholders—*researchers*, *research institutions*, *curators*, *harvesters*, and *developers*—play a governing role in defining the main requirements of a data repository for the management of research outputs. As key metadata providers, *researchers* are responsible for the description of their research data through domain-level records. They are not necessarily knowledgeable in data management practices, but they can provide domain-specific, yet informal descriptions to complement generic metadata, contributing to the much needed data production context that makes it possible for other researchers to reuse the data [36]. As data creators, researchers can play a central role in data deposit by selecting appropriate file formats for their datasets, preparing their structure and packaging them appropriately [75]. *Institutions* are also motivated towards having their data recognized and preserved according to the requirements of funding institutions [85, 154]. In this regard, they value the compliance with metadata standards to make them ready for inclusion in networked environments, therefore increasing their visibility. To make sure that this context is correctly passed along with the data to the preservation stage, *curators* are mainly interested in maintaining data quality and integrity over time. Usually, they are information experts and not knowledgeable in the research domains of the datasets that they curate, so they must work in close collaboration with researchers to produce detailed and compliant metadata records [213].

Considering data dissemination and reuse, *harvesters* can be either individuals looking for specific data or services which index the content of several repositories. These services can make particularly good use of established protocols, such as the OAI-PMH [128], to retrieve metadata from different sources and create an interface to expose the indexed resources. Finally, contributing to the improvement and expansion of these repositories over time, *developers* are concerned with the underlying technologies, but mostly in having extensive APIs to promote integration with other tools.

3.3.5 Architecture

Regarding the architecture of each platform, several important aspects must be considered. From a research institution's point of view, a quick and simple deployment of the selected platform can be an important aspect. There are two main scenarios: either the institution outsources an external service or installs and customizes its own repository, thus supporting the infras-

tructure maintenance costs. Contracting a service provided by a dedicated company, such as Figshare or Zenodo, delegates system maintenance for a timely fee. The service-based approach may not be viable in some scenarios, as some researchers or institutions may be reluctant to deposit their data in a platform outside their control [58]. In this matter, either DSpace, ePrints, CKAN or any Fedora-based solution may offer a better control over the stored data as they can be installed and run completely under the control of the research institution. As open-source solutions, they also have several supporters[17] that contribute to their expansion by developing additional plugins or extensions to meet specific requirements. DSpace, CKAN and Zenodo allow a certain degree of customization to better satisfy the needs of their users: while Zenodo provides parametrization settings such as community-level policies, CKAN, DSpace and Fedora—as open source solutions—can be further customized, with improvements ranging from small interface changes to the development of new data visualization plugins [196, 192]. Due to the complex architecture, DSpace may require a higher level of expertise when dealing with custom features, but its larger supporting community may help to tackle such barriers. The same applies to Fedora as it requires the research institution to choose among different technologies to design and implement the end-user interface, which can exclude it as an option if limited time or budget restrictions apply. We were pleased to see that all packaged platforms provide easy internationalization support. The Zenodo and Figshare services are available in English only, as well as the majority of EUDAT's interfaces—an exception is its B2Share module, which is built on the Invenio platform, which already has internationalization features.

A collaborative environment where teams and groups can manage the deposited resources is becoming increasingly important in the research workflows of many institutions. In this regard, both CKAN and Zenodo provide collaborative tools and allow users to fully manage their group members and policies. As some researchers may want to control the data release dates, ePrints, Zenodo and EUDAT also allow users to specify embargo periods after which data is made available to the community. ePrints and Zenodo are not designed to support real-time collaborative environments where researchers can be producing data and describing them simultaneously, so they can be hard to implement to support dynamic data production environments. It is therefore necessary to create a more dynamic approach to data management, without placing barriers to the researchers, so that they can use these platforms as part of their daily research activities, and while they are still *actively working on the data*—otherwise, only after datasets are finished (no longer in active gathering or processing) will researchers consider depositing them in a platform. Moreover, different researchers may have a different approach to dataset structures and descriptions, and this is very likely to cause difficulties to the deposit-based workflows of these solutions. EUDAT provides a collaborative environment by integrating file management and sharing into the researchers' workflow via a desktop application. This application can automatically synchronize files to one of the environment's modules (B2Drop). After the files are uploaded, they can be used for computation in B2Stage or shared in B2Share to major portals in several research areas. They can also be directly shared to B2Find, the repository module of the EUDAT environment, designed to be a repository for

17 http://ckan.org/instances/
http://registry.duraspace.org/registry/dspace

european researchers to freely share their data. EUDAT's B2Share service is built on the Invenio data management platform. This platform is flexible, available under an open-source license, and compatible with several metadata representations, while still providing a complete API. However, we see how it could be hard to manage and possibly decommission an Invenio platform in the future, since its underlying relational model is complex and very tightly connected to the platforms's code, as we have determined in a past technical analysis [197].

3.3.6 Metadata: a key for preservation

Research data can benefit from domain-level metadata to contextualize their production [222]. While the evaluated platforms have different description requirements upon deposit, most of them lack the support for domain-specific metadata schemas. In this regard DSpace is a notable exception, with its ability to use multiple schemas that can be set up by a system administrator. The same happens with Islandora, which follows the Fedora's support for descriptive metadata, also allowing the creation of tailored metadata forms, provided the correspondent plugin is installed. This can respond to the emergent needs of describing research data with domain-level metadata, a matter that is still to be addressed by many of the remaining platforms. Both Zenodo and Figshare are able to export records that comply with established metadata schemas (Dublin Core and MARC-XML and Dublin Core, respectively), but DSpace goes further by exporting DIPs that include Metadata Encoding and Transmission Standard (METS) metadata records, thus enabling the ingestion of these packages into a long-term preservation workflow. Although CKAN metadata records do not follow any standard schema, the platform allows the inclusion of a dictionary of key-value pairs that can be used, for instance, to record domain-specific metadata as a complement to generic metadata descriptions.

Neither of these platforms natively supports collaborative validation stages where curators and researchers enforce the correct data and metadata structure, but Zenodo allows the users to create a highly curated area within *communities*, as highlighted in the "validation" feature in Table 2. If the policy of a particular *community* specifies manual validation, every deposit will have to be validated by the community *curator*. EUDAT does not support domain-dependent metadata, but it can gather different sets of descriptors when depositing to different projects using B2Share.

For example, when the user that is performing the deposit chooses Global Biodiversity Information Facility (GBIF) as the target project for the new dataset, some pre-defined, biodiversity-related descriptors become available to be filled in as a complement to the generic ones. These domain-specific descriptors can greatly improve upon generic descriptions; the datasets can originate from very specific research domains, thus requiring very specific descriptions to be correctly understood by potential re-users.

Tracking content changes is also an important issue in data management, as datasets are often versioned and dynamic. CKAN provides an auditing trail of each deposited dataset by showing all changes made to it since its deposit. EUDAT deals with the problem of metadata auditing in the same way, because its dataset search and retrieval engine, B2Find, is based on CKAN technology[18], and thus can provide the same auditing trail interface.

18 Please refer to http://eudat.eu/sites/default/files/DaanBroeder.pdf

Table 2: Comparison of the selected research data management platforms

	Feature	DSpace	CKAN	Figshare	Zenodo	ePrints	EUDAT
Architecture	**Deployment**	Installation package or service	Installation package	Service	Service	Installation package or service	Service
	Storage Location	Local or remote	Local or remote	Remote	Remote	Local or remote	Remote
	Maintenance costs	Infrastructure management	Infrastructure management	Monthly fee	Monthly fee	Infrastructure management	Monthly fee
	Open Source	✓	✓	✗	✗	✓	✗
	Customization	✓	✓	✗	Community policies	✓	✗
	Internationalization support	✓	✓	✗	✗	✓	✗
	Embargo	✓	Private Storage	Private Storage	✓	✓	✓
	Content versioning	✗	✓	✗	✗	✓	✓
	Pre-reserving DOI	✓	✗	✓	✓	✓	✓
Metadata & Dissemination	**Exporting schemas**	Any pre-loaded schemas	None	DC	DC, Machine Readable Cataloguing (MARC)-eXtensible Markup Language (XML)	DC, METS, Metadata Object Description Schema (MODS), Data Item Declaration Language (DIDL)	DC, MARC, MARC-XML
	Schema flexibility	Flexible	Flexible	Fixed	Fixed	Fixed	Flexible
	Validation	✓	✗	✗	✓	✓	✓
	Versioning	✗	✓	✗	✗	✓	✓
	OAI-PMH	✓	✗	✓	✓	✓	✓
	Record license specification	✓	✓	✓	✓	✓	✓

Table 3: Key advantages of the evaluated repository platforms

Platform	Key advantages
Figshare	– Gives credit to authors through citations and references – Can export reference to Mendeley, DataCite, RefWorks, Endnote, NLM and ReferenceManager – Records statistics related to citations and shares – Does not require any maintenance
Zenodo	– Allows creating communities to validate submissions – Supports Dublin Core, MARC and MARC-XML for metadata exporting – Can export references to BibTeX, DataCite, DC, EndNote, NLM, RefWorks – Complies with OAI-PMH for data dissemination – Does not require any maintenance – Includes metadata records in the searchable fields
CKAN	– Is open-source and widely supported by the developer community – Features extensive and comprehensive documentation – Allows deep customization of its features – Can be fully under institutions control – Supports unrestricted (non standards-compliant) metadata – Has faceted search with fuzzy-matching – Records datasets change logs and versioning information
DSpace	– Can comply with domain-level metadata schemas – Is open-source and has a wide supporting community – Has an extensive, community maintained documentation – Can be fully under institutions control – Structured metadata representation – Compliant with OAI-PMH
ePrints	– Can maintain records of changes in preservation metadata records – Compliant with OAI-PMH – Compliant with Simple Web-service Offering Repository Deposit (SWORD) for multiple deposit
EUDAT	– Modular approach that provides a variety of services to match local needs – Strong support form European agencies – Integration of several open-source platforms (CKAN, Invenio) – End-to-end workflow for research data management – Majority of features are available for free to european researchers

3.3.7 Interoperability and dissemination

Exposing repository contents to other research platforms can improve both data visibility and reuse [141]. All of the evaluated platforms allow the development of external clients and tools as they already provide their own APIs for exposing metadata records to the outside community, but there are some differences regarding standards compliance. In this matter, only CKAN is not natively compliant with the OAI-PMH protocol [128]. This is a widely-used protocol that promotes interoperability between repositories while also streamlining data dissemination, and is a valuable resource for harvesters to index the contents of the repository [67]. As a government data-centered approach it is understandable that CKAN is missing this compliance, but it can leave research institutions reluctant to its adoption as they can also have interest in getting their datasets cited by the community.

It is interesting to evaluate the ease of discovery by other machines but also how easily humans can find a dataset. All three platforms possess free-text search capabilities, indexing the metadata in dataset records for retrieval purposes. All analyzed platforms provide an "advanced search" feature that is in practice a faceted search. Depending on the platform, it allows users to restrict the results to smaller sets, for instance, from the Engineering domain. This search feature makes it easier for researchers to find the datasets that are from relevant domains and belong to specific collections or similar dataset categories (the concept varies between platforms as they have different organizational structures). As an example, ePrints allows searching through all metadata records, including boolean operators to refine the results as well as full text search for some of the compatible data formats, provided the appropriate plugins are installed. When considering the involved technologies, DSpace stands out as it natively uses Apache

Lucene as it search engine which competes with the Xapian[19] engine used in ePrints, to sort results by relevance.

3.3.8 Platform adoption

As most recent platforms, all the repositories depend on numerous developers to maintain and improve their features. Looking for successful case studies, it is important to assess their impact and comprehensiveness. CKAN has several success cases with government data which are made available to the community, but misses other scenarios related to the management and disclosure of research data. The other platforms—Figshare, Zenodo and DSpace—have research data as their focus. In active use since 2002, DSpace is well known among institutions and researchers for its capabilities to deal with research publications and, more recently, to handle research data. Considering this, DSpace may have better acceptance due to the existence of instances already in place that can ease its management and upgrade. Zenodo is a solution for the long tail of science supported by CERN laboratories, and is regarded as an environment to bring research outputs to a proper digital archive for preservation. It is therefore a strong use case, with all the currently active researchers making use of its features, regardless of their research field.

3.4 DATA STAGING PLATFORMS

Most of the analyzed solutions tend to target the end of the research workflow. They are designed to hold and manage research data outputs after the data production is concluded and the results of their analysis are published. As a consequence, there is an overall lack of support for capturing data during the earlier stages of research activities.

Introducing data management—and metadata production particularly—at an earlier moment in the research workflow increases the chances that a dataset will reach the final stage of this workflow, when it is kept in a long-term preservation environment. The introduction of data deposit and description earlier in the research workflow means that descriptions will already be partially done by the end of data gathering. Also, more detailed and overall better metadata records can be gathered in this way, since data creation contexts are still present in the data creators' memory. Researchers can also reap immediate benefits from their data description, as described datasets can more easily be shared among the members of their research group or with external partners, while exploring them.

Data gathering is very often a collaborative process, so it makes sense to make metadata production collaborative as well. These requirements have been targeted by several research and data management institutions, who have implemented integrated solutions for researchers to manage data not only when it is created, but throughout the entire research workflow.

Researchers are not data management experts, so they need effective tools that allow them to produce adequate standards-compliant metadata records without having to spend too much time learning about those standards. Thus, an important characteristic of an effective solution for collaborative data management is its ease of use by non experts. If these solutions are

19 http://xapian.org/

easy to use and provide both immediate and long-term added value for researchers, they are more likely to be adopted as part of the daily research work. Gradually, this would counteract the idea that data management is a redundant and time-consuming process performed only due to policies enforced by funding institutions, or motivated by uncertain and long-term rewards such as the possibility of others citing the data creator's datasets.

3.4.1 Data management as a routine task

According to the Guidelines on Open Access to Scientific Publications and Research Data in Horizon 2020 [73], a Data Management Plan (DMP) is "a document outlining how the research data collected or generated will be handled during a research project, and after it is completed, describing what data will be collected generated and following what methodology and standards, whether and how this data will be shared and/or made open, and how it will be curated and preserved". The mandatory inclusion of DMPs as part of research funding applications—including those in the Horizon 2020 call—intends to bring Research Data Management (RDM) practices into the research workflow more than ever before. There have been important advancements towards the incorporation of data management practices in the day-to-day activities of researchers, some which can help research groups comply with their DMPs.

In the UK, the DataFlow project [101] was carried out to provide researchers an integrated data management workflow to allow them to store and describe their data safely and easily. The project implemented two components: DataStage and DataBank. DataStage allows researchers standards-based[20] access to shared data storage areas protected by automated backups, as well as a web interface that researchers can use to add metadata to the files that they deposit. The shared storage is accessible directly from the computers that they use for their work through a Windows *mapped drive*. After researchers were ready to deposit a dataset, they could easily package it as a BagIt[21]-compliant ZIP file and send it to DataBank via a SWORD v2 endpoint. DataBank is a repository platform that, besides supporting ingestion via the SWORD v2 protocol, supports DOI registration via DataCite, version control, specification of embargo periods and OAI-PMH compliance to foster further dissemination of the data. File format-related operations—such as correct identification of the format of a file—are handled by existing tools such as JHOVE and DROID [22].

Datorium is a platform for the description and sharing of research data from the social sciences. Realizing the increasing requirements for base data as supplementary material to research publications, its goal is to provide an easy to use platform for researchers to perform autonomous description of their datasets. Metadata is, like in many cases, limited to DC, complemented in this case with some elements taken from GESIS Data Catalogue DBK [5].

MaDAM [174], a web-based data management system targeted at the management of research data in research groups. It provides a user-friendly file explorer, as well as an editor for adding metadata to the entities in the folder structure. The descriptors that can be added to a metadata record are fixed and general-purpose, such as Name, Creator or Comments. The

20 Protocols such as Common Internet File System (CIFS), Secure File Transfer Protocol (SFTP), Secure Shell (SSH), Web-based Distributed Authoring and Versioning (WebDAV) were used

21 BagIt is a specification for the structure of a ZIP file for data and metadata packaging and interchange between data management systems [38]

platform also has an "Archive" function that allows users to send a dataset to eScholar, the University of Manchester's publication and dissemination repository[22].

DASH[23], a data management platform in use at the University of California, incorporates two previous tools: DataUP[24] and DataShare[25]. It does not currently support interoperability protocols for deposit or dissemination of datasets such as OAI-PMH or SWORD, leaving it outside of this comparison. However, it is an open-source project, with its modules currently available on GitHub[26]. It also provides an easy to use interface, indexing by scholarly engines, data identification via DOI and integration with Merritt, an in-house developed long-term repository[27].

HAL is a platform for the deposit, description and dissemination of research datasets. It provides a Wikipedia plugin to modify the layout of Wikipedia pages and directly include links to relevant datasets. This can help researchers find data easily in Wikipedia pages. The metadata that can be added to each dataset is limited to a set of generic, fixed descriptors, whose values can be derived from the content of relevant Wikipedia pages [180].

As a pan-european effort for the creation of an integrated research data management environment, EUDAT [216] also includes a file sharing module called B2Drop. It provides researchers with 20GB of storage for free, and is integrated with other modules such as B2Stage for dataset sharing and staging (running computational operations over the stored data). Complemented with the other modules of the EUDAT ecosystem, it constitutes the entry point of the data into an end-to-end research data management workflow, ranging from the storage of datasets being created to their publishing and later retrieval.

DataHub is a platform that focuses on providing features to allow researchers the versioning datasets as they are created. DataHub is designed to handle large quantities of data, and to store the products of research data, as well as the relationships between datasets and their derivate datasets in a directed acyclic graph [33]. It is made up of two components: a storage and querying layer—Dataset Version Control System—inspired by collaborative version control systems such as Git, and a web-based dataset querying interface—DataHub. The system supports data deposit, querying and visualisation for novice and expert users alike; it allows novice users to use a graphical user interface to build queries over the datasets, while expert users can complement their queries with pieces of Structured Query Language (SQL) that they write themselves to be run over the datasets deposited in the system [34].

3.4.2 The future of collaborative data management

Several interesting concepts have been recently presented as part of an integrated vision for the management of research data within research groups. Some core concepts currently found in social networks are applied to research data management, making it a natural part of the daily activities of researchers [16]. They include a timeline of changes over resources under

22 http://www.escholar.manchester.ac.uk/
23 https://dash.library.ucsc.edu/
24 http://dataup.cdlib.org
25 http://datashare.ucsf.edu/xtf/search
26 http://cdluc3.github.io/dash/
27 http://guides.library.ucsc.edu/datamanagement/publish

the group's control, comments that are linked to those changes, external sharing controlled by the elements of the research group and the ability to track the interactions of external entities over the dataset (citations and "likes", for example). In this design outline, researchers would be able to browse datasets deposited by group members as they are produced, and also run workflows over that data. The continuous recording of not only data but also the translation steps that allow a dataset to be derived from others is a very interesting concept not only from a preservation point of view, but also in research terms, as it safeguards the reproducibility of research findings.

3.5 CONCLUSIONS

The requirements for a data management platform are governed by different stakeholders. Funding institutions push for integration with large platforms, in order to make the results of public investment visible. Researchers require full control over their data, through embargoes, license specifications and overall system security. External systems require the implementation of APIs, harvesting protocols and standardized metadata records so that they can harvest data and metadata. Users interested in reusing the data expect it to be well documented, readily available and free to use. All these requirements have to be balanced in the development and implementation of a data management solution.

We have determined that practically all solutions are compliant with OAI-PMH, which makes it easy for external harvesters (such as repository and dataset directories) to compile their records into large integrated databases. As LOD starts to become more and more prevalent, however, we expect to see the same widespread compatibility on all repository solutions, which will allow richer metadata records to be represented and shared on the semantic web.

There is still a lot of work and research to be carried out to realize OpenAIRE's vision of an environment where project information, funding institution policies, research work and research products are all integrated in a streamlined workflow, enabling reuse and promoting accountability on the results that researchers set out to do when they are granted public funding for research projects.

So far, EUDAT has presents itself as a promising candidate to become the *defacto* RDM platform in the European Union (EU), integrating data management from the start of the research workflow. EUDAT has moved towards a simple interface with researchers, who are not RDM experts. The platform helps them with data storage, description and sharing of their research outputs in a single environment. It has the necessary institutional support too, as it is supported by many prominent institutions in the field, such as the Data Archiving and Networked Services (DANS), Joint Information Systems Committee (JISC) and Conseil Européen pour la Recherche Nucléaire (CERN), among many others.

Dendro, our RDM platform prototype, presents some interesting ideas for the improvement of existing solutions. By proposing the integration of LOD records from the data storage layer up to its flexible and comprehensiveAPI, our platform presents some useful concepts regarding flexibility, both in terms of richer semantics for metadata representation but also in terms of

the integration potential with existing solutions, which can only provided by LOD.

We believe that the future of RDM will follow the path of research itself, in the sense that it will be social and collaborative. As Open Data becomes widespread and data citation assumes a role as important as the publication citation, data staging platforms such as ADMIRAL, EUDAT, Dendro and others will become an integral tool of the research workflow, not only in the sense of scientific production but also in helping researchers get credit from it, as they foster data sharing, data reuse and research reproducibility.

4 PLATFORMS IN SERVICE

Several data management platforms are already in place to support academic and research institutions, entire research domains, or even governments as part of Open Government policies. In this chapter, we start by briefly analysing their purpose of application, following with a study of their main use cases, as well as their technical architecture.

Since there are many data repositories on the web, it can be hard to find the one from a specific domain or which contains certain types of datasets. To cope with this, repository directories have been put in place—these serve as repository catalogues which add metadata not to datasets but to repository records.

The final topic covered in this chapter are dataset directories. These are different from repository directories in the sense that they do not provide lists of platforms but instead index datasets directly, linking to the original record which is harvested from a repository.

4.1 REPOSITORIES

Several institutional repositories for research data are already in place, and we carry out an analysis of their most important features along the present section. The repositories cover several research disciplines, ranging from social sciences, biodiversity and academic datasets, among others. Their purpose, metadata formats, as well as their data indexing and exchange capabilities are analysed.

4.1.1 ICPSR

The Inter-university Consortium for Political and Social Research (ICPSR)[1] is a large research data repository that contains datasets from the domains of political and social sciences. Among the many querying features it provides, this portal allows its users to search for datasets whose tables contain a certain *variable*. This is complemented by the comprehensive *variable catalog* that is available in the platform, which allows users to know the exact meaning of each variable. The portal provides a *variable search* functionality that allows users to search for variables as well, since there are more than 430 thousand variables[2] registered on the website.

The portal uses Data Documentation Initiative (DDI) metadata to annotate its datasets; the metadata can be exported as eXtensible Markup Language (XML) records according to the DDI Codebook 2.1, DDI Codebook 2.5 and

1 https://www.icpsr.umich.edu/

2 As of April 2014: See http://www.icpsr.umich.edu/icpsrweb/ICPSR/ssvd/index.jsp for an updated list.

DDI Lifecycle 3.1 standards. It also provides Dublin Core (DC), Machine Readable Cataloguing (MARC) 21 and DataCite versions of the exported metadata, all in XML format.

Data is made available in a variety of statistics-sofware oriented formats, such as IBM SPSS Statistics Software (SPSS)[3],

4.1.2 NCBI

The National Centre for Biotechnology Information (NCBI) is more than a repository, in the sense that it provides sophisticated tools for the management and use of biology information. It not only stores, indexes and retrieves knowledge about molecular biology, chemistry and genetics, but also provides researchers powerful computational capabilities to perform analysis over datasets stored in the platform. It is also an integrated way for collaboration between researchers in the domain of Biology. The organization that supports the portal also sponsors meetings, lecture series and workshops of researchers from those research domains, and promotes standards for databases, data deposit and exchange, as well as shared biological nomenclature[4].

The NCBI provides implementations of the BLAST algorithm for detection in gene sequences, and is possibly the most used portal for dataset sharing among the researchers from the biology domain. Its deposit procedures are standardized, including the necessary metadata that must be associated to each new dataset.

NCBI illustrates an ideal scenario where the target community of the repository is aware of the advantages of adequately describing each dataset and follows metadata standards. The NCBI platform is also seen not only as a way to share research data but also as a way to get credit for its creation since its adoption among genetics research authors is widespread.

4.1.3 Edinburgh DataShare

The Edinburg DataShare service is a repository for multidisciplinary research datasets produced at the University of Edinburgh. It supports keyword-based search, as well as browsing datasets by *community* and *dataset creator*. In this case, the different communities of the repository correspond to the different schools in the university. Figure 7 shows the layout of the website. The service is powered by DSpace software, which was personalized to have its own institutional look and feel. The platform allows members of the academic institution to perform their own data deposits, and provides a user guide to assist users in the procedure [217].

The Edinburgh DataShare was a result of the Joint Information Systems Committee UK (JISC-UK) DataShare project[5], which also yielded a guide for policy makers in the area of research data management [85].

4.1.4 Data.gov.uk

The Data.gov.uk repository is intended to support Open Government policies in the UK. It is used by the central administration to share data publicly

3 http://www-01.ibm.com/software/analytics/spss/
4 Information available at the NCBI website at http://www.ncbi.nlm.nih.gov/About/glance/ourmission.html
5 Project description available at http://www.disc-uk.org/datashare.html

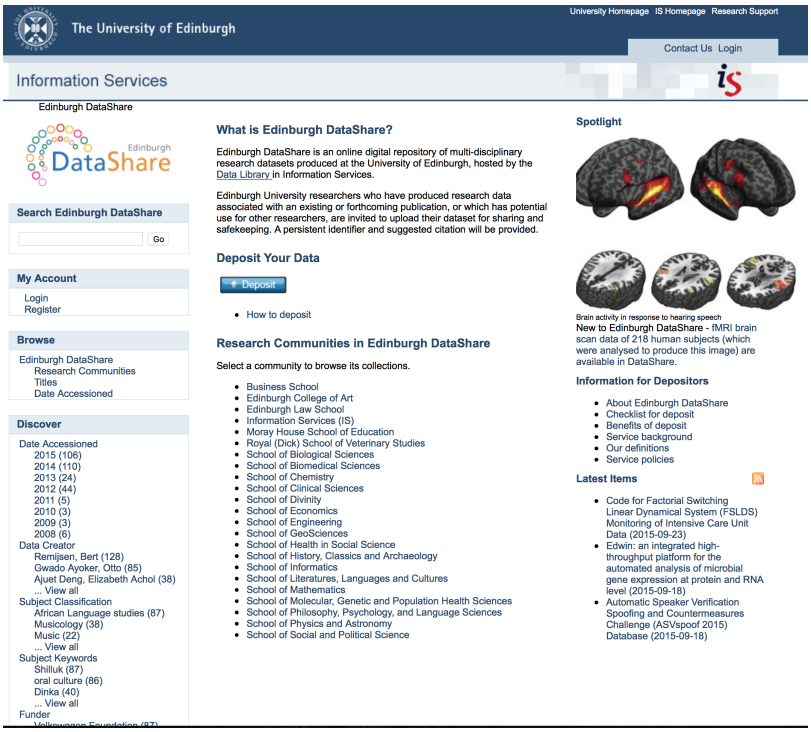


Figure 7: The Edinburgh DataShare portal. Image retrieved in September 2015 from http://datashare.is.ed.ac.uk

in an effort to improve transparency regarding the decisions taken by the government. It provides datasets from all central government departments, giving the public a way to scrutinize policies and the changes brought by them by analyzing historical data.

This repository is powered by a Comprehensive Knowledge Archive Network (CKAN) software platform, and therefore includes a complete Application Programming Interface (API) for data access. This interoperability layer allows external applications to make use of the data stored in the repository and to deposit new datasets in automated way. More than 380 Apps are listed in the repository[6], either using datasets provided by it, or directly connected to it via the API.

4.1.5 DataHub

In spite of the similarity in name between this repository and the DataHub staging platform, this DataHub dataset repository[7] is distinct in its goal and function. Like Data.gov.uk, it is powered by a CKAN instance, but is directly supported by the Open Knowledge Foundation (OKF), and is open to the the deposit of any dataset from any research domain.

DataHub organizes its datasets by license, organisation and *tags*. The organisations are similar to publishing departments for datasets, allowing them to be recorded as the creators of the datasets, instead of allowing only individuals to perform the deposit. For every organisation, there is an administrator that performs validation of dataset deposits within the

6 As of September 2015
7 http://datahub.io

organisation, while giving individual users of the platform authorization for publishing on behalf of that particular organisation.

4.1.6 Dryad

Dryad[8] is a repository for datasets from any research domain, which continued the work started in the DRIADE project [46]. It emphasises open access and links between publications and their base data, providing unique identifiers for datasets, so that they can be cited. It also has its own Application Profile (AP) to guide the production of metadata records which has evolved according to the metadata requirements of the communities in the repository [126].

4.2 REPOSITORY DIRECTORIES

Repository directories—also called *repository hubs or indexes*—are portals that provide organized lists of data repositories, as well as some metadata about each of the indexed repositories. Since data repositories are in many cases targeted at specific research domains, these portals help users find those repositories that contain the data from the domains that they are interested in.

4.2.1 RE3Data

Registry of Research Data Repositories (RE3Data)[9] is the result of the combination of two existing initiatives: *DataBib* and *re3data.org*. The project partners for RE3Data are the GFZ German Research Centre for Geosciences, the Computer and Media Service at the Humboldt-Universität zu Berlin, the Purdue University Libraries and the KIT Library at the Karlsruhe Institute of Technology (KIT). RE3Data is steered by the Research Data Alliance (RDA), which decided on the fusion of the the two existing portals under a common Representational State Transfer (REST) API that enables easy data access and querying.

4.2.2 OpenDOAR

The OpenDOAR (Open Directory of Open Access Repositories) was put in production in 2005. At the time, there were many lists of repositories and open archives in place. However, there was not a single comprehensive or authority list that could compile all of them, while providing a way to search for users to find the right repository—something OpenDOAR aims to solve. OpenDOAR has seen a fast growth since its creation, with 1900 repositories being listed in 2011, but with only a minor percentage of these being dedicated to datasets.

In technical terms, metadata about the repositories registered in the directory is harvested through Open Access Initiative—Protocol for Metadata Harvesting (OAI-PMH). The API provided by OpenDOAR is RESTful and built on XML, whose structure is specified as a Document Type Definition

8 http://datadryad.org/
9 See http://www.re3data.org/

(DTD) specified by the platform itself. API consumer feedback has pointed out that, although it is well designed, the API is "very complex" [119].

4.3 DATASET DIRECTORIES

Dataset directories act as aggregators of datasets. While not hosting the datasets themselves, their role is to harvest the metadata from many repositories and provide a common dataset search interface over the metadata. When a user performs a query using the search functionality of the dataset aggregator, that system will provide a list of datasets that match the query supplied by the user, which are linked to the original (external) record. After viewing the dataset metadata, the user can access the dataset itself by accessing the repository where it is stored.

The dataset aggregator acts therefore as an intermediary between the user and the dataset repositories, providing a single query interface that can make it easier for users to query the contents of multiple repositories using a single search box.

4.3.1 B2Share & B2Find (EUDAT)

EUDAT provides an integrated data management workflow supported by several tools connected by APIs. B2Share, one of these tools, works as a gateway to several prominent repositories that is designed as a user-friendly environment to allow researchers to deposit datasets to several prominent repositories specialized in a variety of research disciplines. B2Share is powered by a modified version of the Invenio software platform.

After datasets are deposited in one of these repositories, the B2Find tool allows users to retrieve them by aggregating the contents of those repositories. The B2Find tool is powered by the CKAN software platform. Figure 9 shows the dataset retrieval interface provided by B2Find.

4.3.2 OpenAIRE

The OpenAIRE project[10] is an European infrastructure that aims to help researchers comply with the European Union (EU) guidelines for Open Access to research results. The OpenAIRE platform collects data and metadata records from various providers, divided into three categories: Literature repositories, Data archives and Current Research Information Systems (CRIS). Research groups are encouraged to connect their CRIS to the infrastructure, allowing for automated harvesting of data and metadata to take place; to enable such interoperability they comply with specific guidelines for interoperability between systems [106].

The Common European Research Information Format (CERIF)-XML format is used for information interchange, which is harvested through OAI-PMH endpoints that must be implemented by the CRIS [105]. All the information collected from the providers is connected in a graph supported on metadata normalised according to controlled vocabularies, with publications being linked to project information metadata through OpenAIRE's knowledge extraction services. User feedback is also included in the process,

10 https://www.openaire.eu/

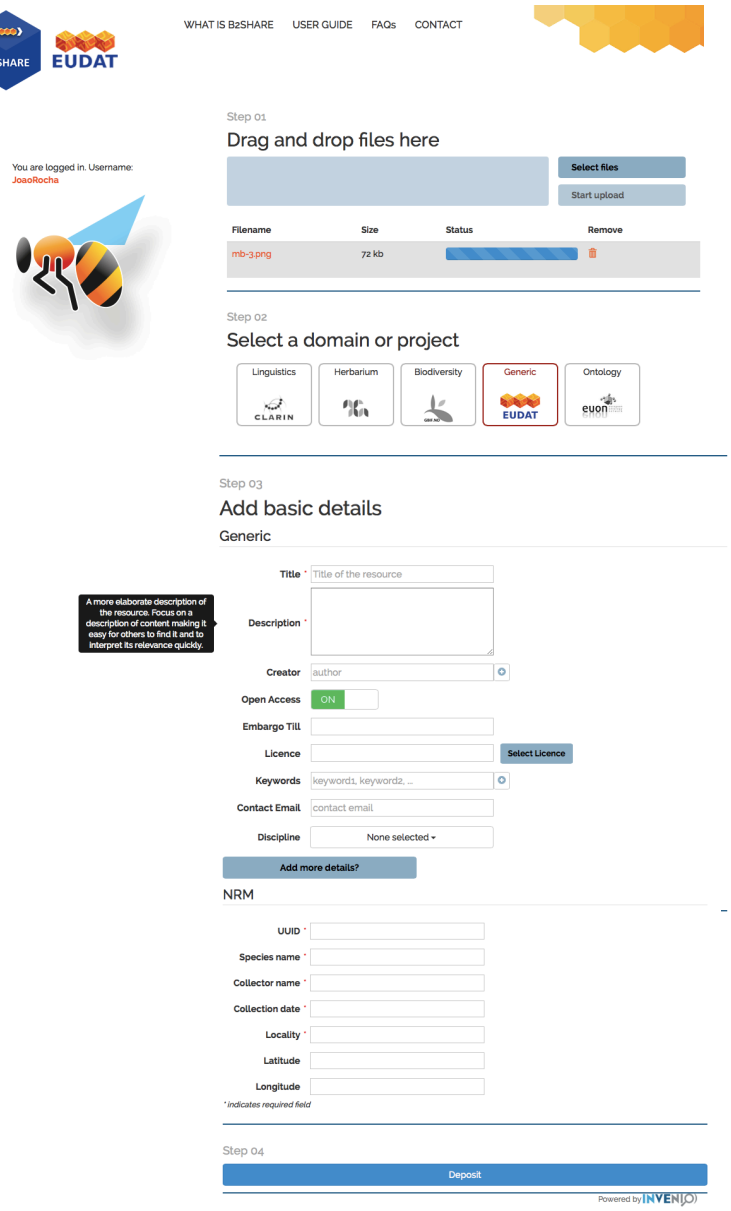


Figure 8: Depositing a dataset using B2Share

Figure 9: Querying for datasets in B2Find

with users being asked to manually claim links between data and publications [177].

4.4 CONCLUSIONS

Various platforms are currently available on the web, allowing for the cataloguing and sharing of many types of datasets. From the now mature repositories of publications to the emerging dataset repositories, all tend to be indexed by directories. Proof of this is the increasingly important role of aggregators in the landscape of data management, which have grown greatly in the last decade—an example is OpenAIRE, which shows sustained growth along that period of time. There is, however, still an unremarkable number of dataset repositories in place when compared to repositories of publications, and they are often focused on specific research disciplines.

Controlled openness is an essential part of the design of data management platforms and all are steered by policy aspects such as specification of licenses for public resources, embargo restrictions or satisfying the requirements of the funding institution. Technical aspects, such the presence of rich APIs or the supported metadata formats can vary greatly across platforms.

The integration between the different solutions in the data management lifecycle is performed at different moments—while some solutions like OpenAIRE harvest data and metadata from CRIS, others such as EUDAT implement self-deposit workflows to try to capture the datasets while they are still in production. In this sense, we are more inclined towards the approach presented by OpenAIRE, as we believe that Research Data Management (RDM) activities should start as early as possible in the research workflow, effectively becoming a natural part of it.

Emerging from publication repositories and relying strongly on the ageing but widely used OAI-PMH protocol, data management platforms are still very much dependent on the DCMI Metadata Terms (DCTERMS) metadata schema for the description of the resources that they manage. As new solutions emerge specifically to target the management of datasets (i.e. EUDAT's B2Find, supported by CKAN), so do the metadata models evolve to encompass the need for more flexible, domain-specific descriptors. APs are thus being encouraged to evolve faster, responding to the evolving needs of the users of the platforms that implement them (e.g. Dryad and its regularly evolving AP).

Support for Linked Open Data (LOD) is also being increasingly backed by repositories, with many of them now exporting their metadata as Resource Description Framework (RDF). In time, it can provide an alternative to OAI-PMH, with support for richer semantics, inference and simplifying the metadata cataloging effort by dispensing (at least in part) the metadata harvesting proces. This will also signal a change for repository and dataset directories, as they will need to adapt to a new reality of LOD-represented records as the norm.

Part II

Recommendation

5 RECOMMENDER SYSTEMS

This chapter is a synthetic analysis of the main approches followed in the development of recommender systems. We start from the item ranking approach, introducing the notion of item and user *features*. Since feature values often need to be normalised, we describe the most common normalization approaches, their importance and potential impact in the performance of the recommender.

Since both the base data of a recommender and its evaluation rely on *user feedback* for data acquisition, we study the difference between explicit and implicit feedback. The importance of both types of feedback in the context of this work is discussed as well.

After the basic notions are introduced, we present the two most widely used categories of recommender systems: content-based and collaborative filtering approaches. Given the importance of Linked Open Data (LOD) in this work, we have also performed a study of past approaches that take advantage of these new sources of information on users and items to improve the effectiveness of recommender systems. The chapter is concluded by a discussion of the different approaches, focused on their ability to perform well in a descriptor recommendation scenario built over a graph of research datasets represented as Linked Data.

5.1 INTRODUCTION

Recommender systems are designed to assist users that lack the experience or competence to evaluate the potentially overwhelming number of alternatives that are made available to them (via a website, for example) [184]. This *information overload* follows the introduction of the concept of *mass customisation* of products. This was a shift from the earlier paradigm of mass production of standardised goods or services to those designed to suit the individual preference of every customer, but on a mass scale [166]. This concept has later been generalised from the products to the entire experience provided to the customer, such as online stores and e-commerce solutions [167]. Recommender systems are an essential part of this concept, enabling this level of customization on a massive scale. Amazon's "customers who bought this also bought..." metaphor is perhaps the most widely known example, but many other e-commerce websites have adopted the concept since its implementation—in technical terms, it is defined as an *item-to-item* collaborative filtering algorithm [136].

Traditionally, recommender systems can be classified as collaborative filtering approaches, content-based or hybrids of the two. Collaborative filtering algorithms are based solely on the preference patterns of the *users*, and not on any information about the *items*. Content-based algorithms rely

on information about *items*, combining it with user ratings to recommend items similar to those that a user has liked in the past. Hybrid approaches try to combine the best features of both collaborative and content-based approaches to improve the performance of the recommender system [20].

According to [184], the first recommender system was Tapestry [81]. Its creators coined the term *collaborative filtering*, which at the time designated an email filtering technique where users collaboratively annotate email messages in order to filter those that are more interesting from within a very large collection of email messages.

Since the main focus of this work is not on recommendation itself but rather on its use to facilitate the process of data description, the overview of recommendation techniques will avoid the algorithmic details of each recommendation technique/method, and rather focus on its applicability to different classes of problems. During our analysis we will discuss which are more adequate to the tackling of our problem and why.

5.2 RANKING ALGORITHMS

In Information Retrieval (IR), ranking is primarily used to select, from among a collection, those that can be potentially relevant to satisfy an information need, expressed through a text query provided by a user in the case of ad-hoc retrieval. Ranking algorithms can also be used for recommendation, in the sense of presenting potentially interesting but previously unknown items to the active user [78].

5.2.1 Weighted sum

Perhaps the simplest ranking algorithm is the weighted sum. In a weighted sum ranking, *features* of the *items* being ranked are multiplied by *weights* specified according to the importance of each feature. The final score of an item is then obtained as the sum of these products. The i_{th} feature of an item, f_i, is a numerical value, while w_i is the corresponding weight.

$$\text{Score}(\text{Item}) = \sum_{i=1}^{n} w_i \cdot f_i \tag{1}$$

The advantages of the weighted sum are its conceptual simplicity and its ease of implementation. The main challenge for the adoption of this approach lies, however, in the definition of the weights. Weights often have to be specified from intuition or experience of those that are designing the system, but a *learning to rank* approach (where the ranking function is learned depending on the response of the users to the retrieved items) can complement this approach. The weight of a feature often expresses its relative importance versus all others, and great care must be taken not to introduce bias in the ranking, which can cause non-relevant items to rise to top positions.

Our approach to descriptor ranking, presented in Chapter 11, is based on the idea of weighted sum, where the score accumulates the contributions of different features.

5.2.2 Document-based rankings

Document-based ranking algorithms take into account the contents and structure of a set of documents in order to rank them against a query specified by an user, which conveys an *information need*. The difference between document-based and link-based rankings is that the former take into account only the structure and contents of the documents, while the latter consider the topology of the graph that interconnects those documents.

Term weighting

Term weighting is an IR technique that is used to represent documents according to the terms contained in them. A simple term weighting schema is based on term frequency and is obtained using the weights of each term are the frequencies with which they occur in the documents (tf). Since this may introduce bias towards those terms that occur many times in a document simply because they are frequent in the collection, it is common to combine it with idf, which grows inversely with the frequency of the term in the entire collection of documents [143]. This yields the statistical measure $\mathtt{tf} \cdot \mathtt{idf}$ which is used to represent how important a term is within a document contained in a collection.

In a hypothetical descriptor recommendation scenario and for the sake of discussion, the role of documents is played by descriptors, which we are recommending to users, while the role of terms can be assigned to the words in their rdf:comment and rdf:label properties. Users could then specify text queries to retrieve descriptors relevant for their domain. Another option to dispense with the manual querying for descriptors could be to calculate the *most important terms* (e.g. through $\mathtt{tf} \cdot \mathtt{idf}$) present in the descriptor instances already present in all the files and folders in the current project. After determining those terms, we could use a lexical database such as WordNet[1] to determine the semantic distance between the rdf:comments of those descriptors and the terms that we determined to be important. The descriptors would then be ranked by their semantic distance and presented to the user.

Cosine similarity

Cosine similarity is a technique frequently used in IR where documents are represented as vectors of terms. The ranking is determined by calculating the similarity between a user query (after its terms are translated into a term vector) and a set of the documents in the collection. The documents are then ordered in descending order of the similarity value, obtained as the cosine of the angle between the query vector and the term vectors of the documents.

For descriptor recommendation, and representing the descriptors as documents, candidate features are the number of times the descriptors were used in different projects, the number of times used in the current project, or other usage counts in different circumstances.

5.2.3 Link-based rankings

Link-based approaches are used to rank pages on the web, as they are hypertext documents that contain links between them. By analysing the topology

1 https://wordnet.princeton.edu/

of the graph of the web, the relative importance of a web page can be calculated. Although we did not adopt a link-based approach in our descriptor ranking, we briefly introduce these algorithms as they can be interesting for future use.

PageRank

PageRank is a widely known ranking algorithm, first proposed by Google in their search engine [161]. It provides a measure of the relative importance of a page by analysing the graph of the web, calculating the probability with which a user randomly navigating through links will arrive at a page. It works on the same principle as citation in research papers, where a citation made in a very often cited paper is more important than a citation present in a paper with a low citation count. The algorithm is run recursively until the scores for the pages converge.

HITS

Hyperlink-Induced Topic Search (HITS), also known the as the *hubs and authorities* algorithm, is a link analysis algorithm used primarily for ranking web pages—although any collection of objects with a link structure of any kind can also be used. It calculates two measures of the importance of a web page: the *hub score* and the *authority score*. In this algorithm, a good *hub* is a page that links to many important pages, or *authorities*. Conversely, a good *authority* is a web page that has may other pages linking to it.

The algorithm calculates these two scores iteratively. The calculation itself can be performed via operations over the adjacency matrix of the pages, which represents the links between them [143].

5.2.4 Learning to rank

Learning to rank designates a class of algorithms where machine learning techniques are used to train the ranking system itself. This means that the ranking algorithm must include parameters that are automatically adjusted to maximize the relevance of the results in the list presented to the users. Learning to rank makes use of optimization techniques such as Gradient Descent and neural networks [42].

5.3 NORMALIZATION METHODS

Normalisation is necessary to offset certain user behaviours which can introduce numerical bias in the ratings that they give, which makes it harder to bring out the real preferences of the users towards items. It can also help to deal with variables with different scales, such as category-based, interval-based or ordinal, which very often need to be normalized before they can be combined. Two main types of behaviours have been identified in the past [115]:

- Shift of average ratings: more *tolerant* users tend to attribute higher ratings than other users.
- Different rating scales: conservative users tend not to rate items using values at the extreme ends of the scale, while more liberal ones tend to only rate items using extreme ratings.

It has been shown that the choice of normalization method can influence the effectiveness of the recommendation algorithm, with changes in evaluation metrics such as Mean Average Error (MAE) resulting from simply from the application of different data normalization techniques [115].

In our recommender (page 134) we use a *min-max* method to normalize the values of the descriptor features. The reasons are its simplicity and the fact that it retains the exact numerical relationships between the feature values present before the normalisation.

5.3.1 Min-max normalization

Min-max normalization is a simple data normalization technique applied in machine learning techniques [90] and in IR works for eliminating bias towards frequent words in a document collection [205]. It is presented as a way to convert feature values between two different scales.

The normalised value of v of a feature A is defined as $v\prime$ (2). In this definition, v denotes the non-normalised rating, max_A denotes the maximum value that A can assume and min_A the minimum value of A. The new_max_A and new_min_a values denote the lower and upper limits of the scale where the normalised values are comprised.

$$v\prime = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{2}$$

The advantages of the min-max normalization are its conceptual simplicity and simple computation (when compared to other methods), but mostly the fact that it applies a linear transformation to the input value. This linear transformation allows it to preserve the exact relationships present in the data, while not introducing any bias [176].

5.3.2 Normalisation by decimal scaling

Han and Kamber [90] define normalization by decimal scaling as adjusting the order of magnitude of the values of a feature A, dividing them by a power of 10, thus moving the decimal point of that attribute. The number of decimal points moved will therefore depend on the maximum value of A. The normalised value is given by Equation 3, where $v\prime$ denotes the normalised value, and v the original value. j is the smallest integer such that $Max(|v\prime|) < 1$.

$$v\prime = \frac{v}{10^j} \tag{3}$$

While it guarantees that the normalised rating ranges from 0 to 1, this measure is very sensitive to the presence of outliers, and is a non-linear transformation. As a consequence, bias is introduced into the normalized ratings if there are outliers, and the relationships between the data are not preserved after the normalization.

5.3.3 Gaussian or z-score normalization

The *Gaussian normalization method* accounts for the biases introduced in the ratings by user behaviours, effectively normalising user ratings by a user's mean rating (thus countering the shift of average ratings) as well as their spread (countering the issue of different rating scales) [102]. The Gaussian normalization method translates absolute values into those values given by a normal distribution. The normalised values, where $\hat{R}_y(x)$ is the normalised rating for item x by user y and $\bar{R}_y$) stands for the average rating for user y are given by: [115]

$$\hat{R}_y(x) = \frac{R_y(x) - \bar{R}_y}{\sqrt{\sum_x (R_y(x) - \bar{R}_y)^2}} \tag{4}$$

5.3.4 Decoupling normalization method

Decoupling normalization, first presented by Rong et. al. [116] and then later benchmarked in a comparison of normalization methods [115], converts ratings into a probability that an item will be favored by an user. The probabilities are calculated according to the following assumptions:

1. "When a large portion of items are rated by a user as no more than category r, items in the rating category r are likely to be favored by the user"—this means that, if r is the highest score that a user ever gives to any item, the items at this top category are the most likely to be favored by the user. The top of the rating scale, so to speak, is established by the user, in a representation of its relative importance versus the lower ratings.

2. "When more items are rated as category r, it becomes less likely for the user to favor items in the category r"—a rating of 4 out of 5 given by an user that rarely rates items as more than 3 should express a stronger preference than when a user commonly gives items that same rating of 4.

In (5), P_y is the normalised score, signifying the probability that the rating R would be given to an item by an user.

$$P_y(R \text{ is favored}) = p_y(\text{Rating} \leqslant r) - p_y(\text{Rating} = r)/2 \tag{5}$$

The two components $p_y(\text{Rating} \leqslant r)$ and $p_y(\text{Rating} = r)$ of the formula are detailed in (6) and (7).

$$p_y(\text{Rating} \leqslant r) = \frac{\text{number of ratings given by the user with value less or equal to r}}{\text{total number or ratings given by the user}} \tag{6}$$

$$p_y(\text{Rating} = r) = \frac{\text{number of r-ratings given by the user}}{\text{total number or ratings given by the user}} \tag{7}$$

5.4 IMPLICIT VERSUS EXPLICIT FEEDBACK

Recommender systems can draw evidence on the preference of users towards certain items in many ways. The two main types of evidence of their preference, or *feedback*, are *implicit feedback* and *explicit feedback*.

5.4.1 Explicit feedback, the most common approach

Users can rate items with varying detail. The simplest way for a user to express preference for an item is through a boolean value (which can represent if the user *likes* or *dislikes* the item). A more sophisticated way is to provide a rating scale whose values can belong to a numeric domain—a scale which can be discrete or continuous.

It is very common to request ratings for items on a discrete 1-5 or 1-10 scale, where 1 means that the user dislikes that item, while the highest rating value expresses the highest preference of the user for that item. Continuous scales have also been presented in the past with positive responses [214], although they are not common practice.

A modification to the scale rating, making it closer to the domain terms instead of numeric, has also been proposed [164]. An example is seen on the popular movie rating website *Rotten Tomatoes*, where users can express positive preference for a movie using a *fresh* rating or negative preference using the *rotten* rating. Another example of these symbolic ratings is the classification of websites on the scale of *hot*, *lukewarm* or *cold*.

Another approach, *multi-criteria ratings*, allows users to rate items along several criteria; more than recording how much a user likes an item, it makes it possible to gather some insight about why the user liked that item [1].

Text comments are the most detailed form of user feedback, but require the most effort on their behalf. They can also be overwhelming for an user reviewing them, since they can be numerous and all require reading and interpretation [187]. Typical examples of this type of feedback are the comments present in Amazon product pages or on eBay listings. As a way to convert these "text-based ratings" into numerical scores, several works in the area of affective computing have been carried out, detecting positive or negative emotions in the comments, for example.

5.4.2 Implicit feedback gathering and scoring

Implicit feedback can be gathered by analysing certain behaviours performed by the user and translating those behaviours into a score. For example, the time users spend reading through a web page or how much they scroll through it can be seen as possible evidence that the user is interested in something in that page. As an example, if the page contains details about a product, it may indicate that the user is interested in purchasing that product, or a product of that type.

Implicit feedback has 4 main characteristics when compared to its explicit counterpart. The first is the inexistence of negative feedback, since there is no way for a user to manifest a dislike for a given recommendation, as there are no rating mechanisms present. Implicit feedback is also noisy, in the sense that it is hard to derive the actual reasons why an user performed a certain action which is used as implicit feedback. Also, in contrast with explicit feedback, where the numerical value of a score indicates a preference, the implicit feedback indicates *confidence*, and thus a higher frequency

of actions or higher browse time is not, *per se*, an indication of a higher preference. Finally, the evaluation of implicit feedback requires appropriate measures; in contrast with systems built on explicit feedback, where standard measures such as MAE can be used to evaluate the performance of the system, in an implicit feedback scenario we must take into account the availability of an item for being picked by the user, or the competition between different items. A typical example is a recommender system for TV shows: when two shows are on at the same time and a user is watching one of them, it does not necessarily mean that the other one is uninteresting, since a user cannot watch two TV shows at the same time [109].

Extracting *User-Item* ratings based on the confidence reading provided by implicit feedback often requires an empirical or numeric study of the importance to assign to each of the different implicit feedback *behaviors*. Several of these behaviors have been classified into categories like *Examination*, *Retention* and *Reference* behaviors [158]. Examples of Examination behaviors include the selection of a given item by the user (wanting to know more about it, for example), purchasing an item, accessing an item repeatedly or listening to a song. In those cases where the interaction with the system is so limited that not even these behaviors are available in sufficient numbers, the duration of a visit to a given resource—the time an user spends viewing that resource on a web page, for example—can also be discretized by classifying according to normalized intervals, to which different *implicit ratings* can be assigned [212]. Examples of Retention behaviors include "saving an item for later" (with or without adding annotations, which can later be considered a useful source of recommendation context), as well as printing or deleting it. Reference behaviors can be, for example, linking a resource to another or copying and pasting information from one to another.

Implicit feedback-based approaches have been considered to be "substantially harder" to implement than those scenarios where explicit feedback is available [215], and implicit feedback can be considered to be less precise than its explicit counterpart, since external biases can be introduced in the resulting data by various reasons. A study of click-through behaviour over Google search results [117] proved that the order by which Google ranked its query results can introduce a so-called "trust bias" by leading users into clicking results ranked higher. Another bias, called "quality bias" introduced by the quality of the abstract presented below each search result (when compared to all the others in that particular result list) was also shown to lead the user into clicking that particular result. The authors concluded that implicit feedback has to be analysed in conjunction with its context—for example, when analysing the feedback provided by a selection of a result from a list of results in a Google search, one must consider the order of presentation at the time as well as the ranking of the abstract of the selected item, when compared to the ranking of the abstracts of the other search hits.

5.5 CONTENT-BASED RECOMMENDER SYSTEMS

Content-based recommender systems take past information about which items an user has liked in the past and try to recommend similar items to that user [20]. The technique has its roots in IR studies, where the text content of documents was modeled as a bag of terms and compared against

the bag of terms extracted from a user profile. The result is a system capable of recommending users a set of documents similar to their profile.

5.5.1 Item features

In order to calculate the similarity between items, it is necessary to translate certain characteristics of those items into comparable numeric values. The extraction of features from the items is perhaps the most challenging part of building a content-based recommender system, since it requires good knowledge of the domain in question. As an example, let us analyse a recommender for movies. Some features might be the genre, the actors or the director (boolean features). In the case of a restaurant recommender, one could select the price as a feature (numeric scale), the location (geographical coordinates), types of food served (vegetarian, steak—boolean features). A way to calculate similarity between two items is the application of a cosine similarity measure, where the values of the features are the dimensions of a vector representing each item.

It is not easy to gather different types of features and reliably combine them in such a way that a single measure of distance can be calculated. This might be solved, for example, by grouping the features of a given category (booleans, coordinates, numeric scales) into several User-Item matrices. After calculating the distances between items based on each matrix, the results might be weighted in order to come up with a final distance measure. The challenge is how to determine the weights for each component—this often depends on the domain where the recommender system is being applied.

5.5.2 Advantages and disadvantages

The advantages of content-based recommender systems when compared to collaborative filtering approaches are *user independence*, *transparency* and ability to deal with the *new item problem*. User independence means that the system relies only on the ratings provided by the active user; by contrast, collaborative approaches require the presence of ratings of other users in order to detect the active user's *neighbours*. This difficulty in determining any neighbours of the active user will prevent the collaborative algorithm from providing any recommendations at all, which constitutes one of the forms of the cold-start problem.

Regarding transparency, since the recommendations emerge from similarities between the recommended items and items that the active user has liked in the past, it is easy to justify those recommendations. In a collaborative filtering approach, recommendations are calculated by calculating similarities between the active user and others, and it may provide no insight to the active user on which users are similar to them.

The *new item* problem (recommending items that have not yet been rated by any user) is also tackled by content-based recommender systems. Since they do not rely only on the ratings given by users, new items can be recommended by similarity to others already rated, instead of having to wait until a significant number of users have rated the new item.

The disadvantages of this approach are related to the need to know the domain of the recommender system in order to select those features that can set apart some items from others. While in some applications textual analysis may be enough to differentiate items (such as in web page recommen-

dation), in other cases, more specific features should be derived—sentiment analysis of poetry [187] or famous quotes are some examples.

5.5.3 Applying content-based recommender systems to descriptor selection

In Dendro, the Reserach Data Management platform described in Chapter 8, all descriptors have textual descriptions (rdf:label and rdf:comment) added to them when they are defined in their source ontologies. A possible scenario for building a pure content-based recommender is be to ask users to provide some information about their research domain and then extract the most important terms in those descriptions (for example, through a term frequency analysis). The same text analysis could also be performed over the text contents of the rdf:comments of the descriptors, and the most important terms could be considered the *features* of the descriptor.

The main issue with this approach is that the short texts that annotate the descriptors may not have any textual similarity to a text entered by the user to describe their own work. To try to cope with this, we could add descriptions to the ontologies in addition to those of descriptors, and compare those descriptions of the ontologies to the textual profiles of the users. The descriptions of the ontologies would likely be more related to the research domain from where they emerged, giving another source of similarity between descriptors. In other words, our items (descriptors) would have two descriptions (parent ontology description and descriptor description); these could then be analysed and the most important terms extracted using $\mathtt{tf} \cdot \mathtt{idf}$ scoring, for example, and the important terms would be used as features.

This scenario, while apparently straightforward, relies heavily on the descriptions that the researchers provide of their own work, in order to match descriptors with researchers. These descriptions should contain words similar to those in the descriptions of the descriptors. Similarly, descriptor-descriptor similarities would be determined from the most important terms in their descriptions.

From our ontology modeling experience with researchers, we can say that there is no guarantee that these descriptions contain similar terms or, from another perspective, that the presence of the same terms in a description effectively represents a similarity between the descriptors.

5.6 COLLABORATIVE FILTERING

The term *collaborative filtering* has been adopted to designate a class of recommendation algorithms that recommend to the active user the items that other users with similar tastes rated highly in the past. It is a process that tries to mimic the "word-of-mouth" phenomenon, and has hence been called "people-to-people" correlation [199]. Collaborative filtering depends only on the ratings given by the users to the items that can be suggested, making this approach more general than content-based filtering (which requires the selection of item features that often suit the particular domain of application) or ad-hoc information retrieval [208, 149].

Collaborative filtering algorithms can be classified into two sub-categories: Memory-based and Model-based. The memory-based approaches resort to a database of ratings given by users over items to produce recommendations, while the model-based approaches take advantage of a database of user

ratings to estimate or learn a model, which can then be used to predict what the ratings of items by an user might be [39].

5.6.1 Neighbourhood-based recommendation models

A common type of collaborative filtering algorithms are neighbourhood-based recommendation models; these enjoy great popularity due to their simplicity, efficiency and effectiveness [66]. This sort of model represents its base information as an User-Item matrix, where each user is represented by a vector of ratings given to some of the items present in the system.

Let us analyse a very simple example to illustrate two basic notions of a neighbourhood-based recommendation model: *similarity* and *neighbourhood*. If user A likes Rock and Blues music, user B likes Rock and Pop music, and user C likes Country and Jazz music, a *similarity* can be calculated between A and B, based on the fact that both share a taste for Rock. After defining a similarity measure, we can calculate the range of users that will be considered as similar to the current user. This set of similar users are called the *neighborhood* of that user. In this case, the neighborhood of A would be B and the neighborhood of B would be A (thus excluding C, since it has no common interests with either A nor B). Now, if A is the current user, the interests of B that A does not already like (in this case, Pop music) would be recommended to A, and vice-versa. This is the basis of collaborative filtering. *Similarity* is therefore the criteria for the inclusion of a user in a *neighborhood*. To determine the similarity between users, statistical measures such as cosine similarity, Pearson correlation or Spearman rank correlations can be used [70].

In the case of our descriptor recommendation, we could calculate a similarity measure based on the descriptors that were used. If a user A fills in a series of descriptors and another user B also fills in another set, the similarity could be calculated from the number of times that both users filled in the same descriptor (or simply a boolean value stating if they have filled in that descriptor or not).

5.6.2 The cold-start problem

The so-called cold-start problem is defined as the inability of the system to provide recommendations to users that have not provided any ratings to any items in the system. The term cold-start problem can also be applied to the items, designating the inability to recommend items that have not been rated by any user in the system [200]. When applied in this second context, the term *cold-start* is also designated as the *first-rater problem* [149].

This *cold-start problem* can be pictured in our context if we analyse the diversity of research domains and the specific nature of each descriptor. Since there are many descriptors to choose from and some of them are only relevant for a small number of users at first, it is hard to obtain the first ratings that are necessary for the model to begin producing recommendations for a researcher that just started to use the system, or for new descriptors to begin being recommended to existing users. In our case we expect, at the start, a low number of users when compared to the number of items (possible descriptors) that they would have to rate, so it was clear that the cold-start problem would be present—this led us to prefer a descriptor ranking approach in detriment of a collaborative filtering approach one.

5.6.3 Applying the technique to descriptor selection

A straightforward application of a collaborative filtering algorithm for descriptor recommendation might be to consider researchers as the users and descriptors as the items. After that, we might ask our users to rate descriptors on a scale, say from 1 to 5; similarities between users might then be calculated through the descriptors that they rated.

The number of people in the panel of researchers that agreed to take part in our experiments as test users was small and with limited time availability. If we had designed a system using a collaborative filtering algorithm, we: 1. would have to explicitly request users to rate several descriptors somewhere throughout their data description, introducing a change in their normal description behaviour and consuming more of their time, and 2. would risk not being able to provide any actual recommendations due to the cold-start problem.

5.7 LINKED DATA IN RECOMMENDATION

Linked data can be a good source of data to support a recommender system. DBpedia—a prime source of Linked Data–has been used in the past for music recommendation, with good results when compared to alternatives such as last.fm [163]. Past work has introduced the notion of Linked Data Semantic Distance (LDSD), which is used to compute the distance between two resources published as Linked Data [163]. This measure can be used to calculate a semantic distance between two artists, for example, allowing the system to recommend artists similar to others that the user liked in the past. In this approach, linked data is also used to provide more transparency to the end users, because the recommendation system can give better explanations as to why a certain artist was recommended to an user. To do this, the recommender analyses the types of links (i.e. the properties linking the resources). For example, if two artists share the same dbo:voiceType[2], the user interface will present that fact as part of the explanation for recommending an artist (the artists share the same voice type). Large public knowledge bases such as Freebase and DBPedia also made it possible to build recommender systems to support long-term preservation workflows, using a recommender system to support file format recognition in addition to existing solutions such as DROID or JHOVE [82].

LOD has been used to enhance collaborative filtering, reducing the sparsity in the User-Item matrix which is the main cause of cold-start problems (new-user or new-item), affecting this type of recommender systems [96]. The expensive acquisition of user ratings was complemented with cheaper data gathering alternatives: LOD sources such as DBTune, a knowledge base derived from MySpace data, were used to gather information about events, users and artists in a music recommendation scenario.

Information from LOD sources was also used to complemented a hybrid recommender that relied on both Collaborative Filtering and Content-Based algorithms [123]. This work gathered data from three large public event directories (Last.fm, Eventful and Upcoming) and converted it into LOD to recommend events to users based on their past event attendances, as well as the similarity between events. To calculate the similarity between the events

2 Property defined in the DBpedia ontology, and available at http://dbpedia.org/ontology/voiceType

and suit them to the profile of each user, a topic detection model—Latent Dirichlet Allocation (LDA)—was used in order to determine the characteristics of the events that certain groups of users liked the most. Factors such as the geographical distance between events that the users attended were also taken into account as event features.

Learning to rank approaches can also benefit from information gathered from LOD sources [160]. In this case, movie and music recommendation was based on implicit feedback by the users and leveraged DBpedia to calculate path-based features. Those features were then used to find items similar to those over which an user has expressed interest. The presence of more paths between users and artists would indicate a stronger preference of the user for that artist. Different types of paths were present, which combined user ratings (*likes*) with content-based similarity relationships gathered from common DBpedia properties. After calculating the path-based features, a learning to rank algorithm was put in place to devise a ranking function for a *top-N* recommendation task. This task differs from other recommender systems in the sense that it does not attempt to predict the ratings that a user would give to the different items, but instead tries to find a few specific items that might be relevant to the user.

We applied LOD to descriptor recommendation as well in past experiments. After analysing the text contents of a document, we extracted the most important terms according to a $\mathtt{tf} \cdot \mathtt{idf}$ measure and queried DBpedia using those terms. The properties of the results returned were then counted, and the top-N most frequent properties were presented as possible descriptors that could be filled in for producing a metadata record for the document in question [194].

5.8 CONCLUSIONS

From the proposed recommendation algorithms, we adopted a weighted sum ranking for our descriptor recommender. We also opted for implicit feedback in our descriptor recommender, because we wanted to make the training of the recommender as unobtrusive as possible for the data description workflow, thus creating a scenario that more closely resembles a production-ready approach. Implicit feedback signals are also quicker to gather since they do not require the user to rate descriptors; this helps us circumvent the serious cold start problem present in our system. This made the model more complicated because we had to devise a scoring formula to convert implicit feedback signals into comparable feature values. For the sake of analysis and discussion, we now discuss the explicit feedback functionality that we implemented to complement the implicit feedback.

From the short analysis of the state of the art on using LOD to complement recommendation approaches, we can say that the emphasis of the research in this topic resides in using the information available in LOD knowledge bases to help deal with common problems of recommender systems, such as the cold-start problem, while improving precision and recall. LOD can also be used to provide human-understandable explanations for the recommendations provided by these systems.

6 EVALUATION OF RECOMMENDER SYSTEMS

This chapter presents a brief overview on the most common recommender system evaluation techniques. We present the main differences between them and the circumstances in which they can be applied. We refer to some characteristics of our own descriptor recommendation scenario, to help frame the reasons why we adopted a user study in detriment of the other possible experiment types.

In this overview, we consider the three main types of evaluation experiments for assessing the effectiveness of a recommender system: offline studies, user studies and online experiments. We also mention some evaluation metrics commonly found in Information Retrieval (IR), which we adapted to complement evaluation of our recommender system.

6.1 OFFLINE STUDIES

An offline experiment is performed using pre-collected user ratings, usually reference datasets already adopted in the community (MovieLens[1] is a widely-known example, as well as the datasets from the yearly challenges presented in the Association for Computing Machinery (ACM) conference on recommender systems[2]).

The cheapest in terms of effort, time and resources, offline studies enable researchers to experiment with many different evaluation versions of their recommender algorithm quickly and in a controlled environment, in order to quickly narrow down the possible algorithms to a few candidates. Offline experiments are also useful to tune the parameters of a recommendation algorithm prior to testing it with real users in a more expensive type of experiment such as a user study.

The offline study has the shortcomings of only answering a small set of questions, typically regarding the prediction power of an algorithm. When the recommendation algorithm is put in production, users may express different behaviours in response to the recommended items [202].

In our case, we tried to perform offline studies by generating synthetic data, but without good results. Early prototypes would not work well with synthetic data, perhaps due to the inadequate nature of our synthetic datasets. Synthetic datasets are time-consuming and difficult to generate, because it is not easy to represent the complex correlations present in real datasets [6], while the synthetic data may be "unfair" to different algo-

1 http://grouplens.org/datasets/movielens/
2 http://www.recsyswiki.com/wiki/Recommender_Systems_Challenge

rithms [97]. Without a known baseline, reference dataset, or synthetic data, we had to establish a baseline through a user study.

6.2 USER STUDIES

User studies are performed by small groups of users that work with a realistic system to simulate a real-world task where recommendation is present. Throughout the experiment, users are asked to report on the experience, either through quantitative or qualitative information. These studies are carried out in a controlled environment, where users can be asked to perform a set of tasks while they are monitored.

Questionnaires can be used as a way to gather feedback on the relevance of the recommended items, or purpose-built applications can be built in order to present possible recommendations and have the users assess their relevance. Examples of quantitative information are the percentage of the tasks that are completed successfully or the time that each task takes to be completed; qualitative information includes feedback on whether or the user liked the User Interface (UI) or whether the user found the tasks easy or hard. The main difference between the *user study* and the *online experiment* is that users are aware that they are being evaluated during the time they are performing their tasks. This can introduce biases in the results since the users may not perform their tasks the same way when they may feel that their behaviour is being scrutinised [203].

In the design of our evaluation experiments, we have drawn inspiration from past works where the number of ratings and test users was small [212, 117, 84]. In these studies, the size of the user group that participated in the studies ranged from 15 to 34 users.

There is a difference between the definition of user study presented here and the evaluation experiments described in Chapters 10 and 12. The participants in our user study were not aware of the fact that we were evaluating their "performance" regarding descriptor selection. The experiments were presented as a user test of Dendro, our data management platform, instead of focusing on descriptor selection.

The reasons for adopting a user study in our experiment were that it allows for a close proximity to the users, enabling us to gather personalised feedback, introduce basic data management concepts and teach them how to use the platform. We also worked closely with researchers to define the descriptors relevant to their domain (as ontologies) and accompanied them while they described their data.

6.3 ONLINE EXPERIMENTS

Online experiments are perhaps the most trustworthy experiment regarding the quality of the gathered results, but are also the most expensive to perform, in terms of time and users involved. They consist of a large-scale scenario where users perform their tasks using a system. Meanwhile, their interaction is monitored without them being aware that they are being subjected to a test [203]. These interactions can be either *explicit ratings* or *implicit ratings* that, through various analysis processes are translated into *User-Item* ratings.

Our experiment, in particular, shared a common trait with an online study: we never mentioned to users that their actions would be taken into account for recommending descriptors, or that there was a recommender system in operation at all. Also, they were not encouraged to interact in any particular way with the system after the initial tutorial was shown, and our role was limited to explaining a particular function of the system if the user had any doubts about how it worked. This way, we tried to simulate a realistic data description scenario in the same way it would occur in the daily activities of a research team.

6.4 COMBINING USER INTERFACE AND INFORMATION RETRIEVAL METRICS

The particular nature of the descriptor recommendation problem calls for the combination of IR and User Interface (UI) evaluation metrics.

In our evaluation, we have adopted some performance metrics from conventional ones such as *precision* and *recall*. Precision can be implicitly determined by the number of clicks on the results presented in the list of descriptors produced by the Dendro interface (which are descriptor selections by the user). Precision@n, or P@n (the percentage of relevant documents in a results list of length n) is particularly interesting. In this case, the descriptor selection list can presents a small number of descriptors, thus our n is equally small ($n = 6$ in our experiments). If we consider the size of the list as the value of n, then the P@n value can be calculated by dividing the number of descriptors selected by the user from a list of n descriptors by n.

6.5 CONCLUSIONS

In this chapter we have briefly presented some common techniques for evaluating the performance of recommended systems.

Describing research datasets requires expert knowledge in their research domains, which greatly narrows down the number of users that can be included in the evaluation. After analysing the possibilities for the evaluation of our proposed approach, we have opted for a user study. The reasons were the number of users that we expected would participate in the experiment (which would be too small to support a large-scale online study) as well as the very specific nature of the data and its description. The absence of a reference dataset and results to serve as a baseline also made us exclude an offline study, but we hope that further research on improving the recommender system can be performed using our dataset[3]. User studies are also more adequate in those cases where users need to be taught the basics about the task (in this case, research data management). In other situations such as music recommendation, users can be expected to carry out simple evaluation tasks without any support, allowing the experiment designers to perform large-scale online studies—through *crowdsourcing*, for example.

3 Base data available at http://dendro.fe.up.pt/demo

Part III

Dendro

7 REQUIREMENTS

This chapter covers the process of requirements gathering for the development of Dendro, a prototype Research Data Management (RDM) platform designed and built during this work to integrate data management in the research workflow, and which served as the basis for our evaluation experiments. It starts with the identification of the stakeholders in a research data management process, as well as their most common needs in that context. After analysing the role of the stakeholders in the RDM workflow, we outline the research workflow commonly followed at research institutions who manage their data (which usually separates data production from its description). We then compare it to our approach, which implements collaborative data management from the moment the data is created.

As high-quality metadata is an essential part of our RDM workflow, we explain the changes that need to be implemented in the Dendro platform in order to accomodate domain-specific metadata. We compare our data model with those present in other common data management platforms and wikis, demonstrating that only the implementation of a platform with a data model such as ours (and not reusing an existing platform) could allow us to satisfy all the gathered requirements.

7.1 INTRODUCTION

Just as institutional repositories have been brought about by such powerful tools as DSpace, EPrints and Fedora, data repositories require tools that simultaneously engage researchers and are appealing to data curators. Many such tools are being experimented, but the spectrum of requirements is so vast that no single tool is yet handling them all [13]. We have therefore addressed research data management from the point of view of the researchers, identifying their main requirements and devising or adapting tools to make the data preparation task easier and more targeted at publication [196, 192].

It has been shown that the absence of timely description from the start of data production can yield lackluster descriptions [145], and also that research data management should start as soon as possible in the research workflow [21]. However, most research data management platforms like Comprehensive Knowledge Archive Network (CKAN), Zenodo or Dryad are designed for publishing *finished* datasets that can be cited. This *a posteriori* data management timing yields very high-quality and highly-selected datasets, but many interesting datasets are likely to miss the opportunity to be published. Empty dataset archives and repositories are still commonplace [155, 36].

Researchers often leave their teams after publishing their papers, and without having the chance to describe their datasets. A possible solution

would be to have data curators accompany the research workflow—however, small research groups may struggle to keep up with the description demands posed by the existing datasets, and it is not realistic to expect researchers to spend much time in data description activities. A possible compromise scenario is to support researchers in the description of their data as they produce them, postponing curator intervention until later in the workflow.

Data management is usually performed "a posteriori"—that is, at the end of the research workflow and after the results have been published. This places research data at risk, since researchers often move on to new projects as funding ends, making it hard to obtain timely and comprehensive metadata for these large sets of research products. To prevent this, research data management process should start as early as possible [145, 75], ideally as researchers have knowledge of their data production context and are actively producing their datasets [21]. Unfortunately, it is much more demanding to produce metadata good enough to support sharing and reuse than it is to produce metadata for personal cataloguing and sporadic sharing [36]. It is a fact, however, that researchers often feel the need to produce basic metadata (such as README files) for personal management [114, 193] as they produce datasets, and that they share this metadata within their research groups in an ad-hoc manner [144].

Several data management projects focus on supporting collaboration within research groups and making daily data management activities easier. The resulting tools are therefore entry points through which the datasets can enter a preservation workflow [101, 204]. These solutions focus on providing easy-to-use shared storage spaces with regular automated backups, connected to a data repository. The main objectives are to capture data as early as possible and leave detailed description for later (*curation by addition*). In these cases, only a minimal set of metadata is required upon initial submission, leaving the decision to enrich the metadata to the researcher and/or curator.

Current data management platforms often limit the metadata that can be added to a dataset to generic descriptors (e.g. Dublin Core (DC)) or a pre-existent set of descriptors that depositors are asked to fill in at the time of deposit. CKAN [159] is an exception, as it allows an additional set of arbitrary metadata to be added to a deposited datasets, in the form of *ad-hoc* key-value pairs, entered in text fields at the time of dataset deposit. This allows domain-specific metadata to be recorded, although without any pre-defined meaning or standards-compliance.

In this chapter we provide an overview of Dendro, our prototype data management platform [190, 189]. Dendro is an ontology-based [197] staging platform for describing and publishing datasets, designed to suit the needs of small research groups in the long-tail of science. The platform is built on the requirements gathered from our contacts with a panel of researchers that accompanied our work, and as such was designed to suit the most common needs expressed by all the groups, instead of being tailored towards the needs of a specific research group.

This need for a cross-domain research data management platform means that we do not know, at the time of database design and modeling, which descriptors will be added to each data resource during the description of the latter. Hence, this calls for a data model that can accomodate unknown and variable attributes for each resource that is described; these attributes can also vary with time, as descriptions are modified by the researchers. To cope with these constraints and still keep the data model easy to understand and

query, we opted for a graph model for the data storage and querying layer of Dendro. We will go into more detail regarding the reasons for this choice by presenting a comparison between several data management solutions built on relational databases, as well as the (arguably) most popular semantic wiki platform, Semantic MediaWiki (SWM).

This chapter is based on the following publications [188, 195, 197]:

- Rocha, J., Barbosa, J., Gouveia, M., Ribeiro, C., & Correia Lopes, J. (2013). UPBox and DataNotes: a collaborative data management environment for the long tail of research data. iPres 2013 Conference Proceedings.
- Rocha, J., Ribeiro, C., & Correia Lopes, J. (2013). UPBox e DataNotes: um ambiente de suporte à gestão colaborativa de dados científicos. InCID: Revista de Ciência da Informação e Documentação, 4(2), Universidade de São Paulo-Ribeirão.
- Rocha, J., Ribeiro, C., & Correia Lopes, J. (2014). Ontology-based multi-domain metadata for research data management using triple stores. In Proceedings of the 18th International Database Engineering & Applications Symposium.

7.2 STAKEHOLDER REQUIREMENTS

Metadata creation involves several tradeoffs, most notably the one between the need for detailed descriptions and the effort required from researchers in order to produce them. The data management needs of researchers and research groups in different areas have been studied in the past [204, 101]; among these, we focus on the requirements for metadata production during the research workflow, which involve mainly the researchers.

1. **Detailed description as a requirement for reuse**
   Research data reuse relies on rich metadata, since researchers need to be aware of the data's production context. Experimental methods, locations, creation or modification dates, substances under study or specific instruments and tools used are only some of the relevant information about a dataset that can be relevant for dataset interpretation. These descriptors can be generic, as the ones specified in the DCMI Metadata Terms (DCTERMS) metadata schema, or domain-specific (as the description of studied variables in Inter-university Consortium for Political and Social Research (ICPSR)[1] datasets).

2. **Fast deposit**
   Sometimes researchers can lose track of some datasets, as well as of their production contexts on certain events—the occurrence of changes in the research teams or the conclusion of projects being prevalent examples. Barriers to initial deposit should be eliminated, namely initial metadata requirements. No researcher should have to fill in a lengthy metadata sheet to submit data (at least at a first moment).

3. **Researcher-driven metadata**
   If metadata descriptions are to be richer (i.e. domain-dependent),

1 ICPSR http://www.icpsr.umich.edu/ is a web portal designed for the deposit and sharing of social sciences datasets

metadata descriptors should be initially selected by the researchers, as they know which descriptors are relevant for other researchers in the field and how to fill in their values. These descriptors should, however, be designed by data curators—this way, both metadata standards compliance and domain relevancy requirements can be met.

From an interface design point of view, these goals call for a simplified and easy-to-use interface, since the target users would be the researchers themselves. Data management also involves some very sensitive aspects such as handling private data, so a comprehensive system for handling data access should be put in place. It should be easy for a special group of users in the research group to manage who should have access to the project.

When the user is allowed to access the data and metadata, they should be easily accessible by both humans and machines, calling for an interoperability layer built on existing standards. From an interoperability point of view, Linked Open Data (LOD) is a prime candidate for the representation of metadata records and has been considered from the start in the development of Dendro.

From a database design point of view, the need for diverse metadata means that the system must allow entities to have different attributes, which are *unknown at the time of modelling* and are *versioned* in time. At the same time, there is often implicit meaning in the file and folder *hierarchies* managed by researchers in their everyday work, so these should be carried over to the data management platform. For a relational system this poses several technical challenges regarding the number of tables involved (and consequently heavy JOIN operations) and the complexity of the queries that have to be written. Relational models also struggle to represent entities with unknown and variable attributes (at the time of modeling) and are unable to cope with their different types (text, integers, dates, etc.).

Complexity is undesirable from a preservation point of view; a relational schema becomes arguably more *closed* as complexity increases, since more effort goes into understanding the data model in order to successfully query the database. From a preservation point of view, simplicity is a goal to strive for because it delays obsolescence. The more people can contribute to a software solution, the more easily it can be kept up to date [57], thus delaying obsolescence.

Due to their flexible structure, graph representations are better than relational databases at representing resources with varying attributes. In our case, this is applied to the representation of metadata for research datasets. Moreover, relational schemas rely on external documentation to expose the meaning of the tables and columns in the schema. In the absence of such documentation, when the developers disperse so does the knowledge of the relational model [8, 181]. In contrast, by reusing ontology concepts the graph can become a self-documented model, as the semantics of all the properties and classes instantiated in it will be publicly available within their originating ontologies via *Annotation Properties* [147]—another benefit to consider when designing a system for preservation.

7.2.1 Full linked open data compliance

Easy and standards-compliant dissemination is essential to the retrieval and reuse of research datasets. Representing dataset metadata records as Linked Open Data (LOD) allows both humans and machines to interpret the meaning of the metadata values present in a record, as well of the descriptors

themselves. This is because the LOD guidelines provide a standard syntax for representing the records (typically through Resource Description Framework (RDF) or Web Ontology Language (OWL)), and a standard way to query and retrieve them (SPARQL Protocol and RDF Query Language (SPARQL)). The semantics of the represented entities and of the relationships between them are represented as well.

When designing Dendro, we started from the assumption that all the data stored in the platform, as well as the associated metadata, would have to be easily accessible from outside systems—without compromising access control and permissions management. In line the with the LOD guidelines, the system should be able to respond differently depending on the request sent by a client. For example, if an external system de-references a record, Dendro should respond with an RDF representation, while a human using a browser should receive an Hypertext Markup Language (HTML) representation of the same record. This is achieved through Content negotiation between the client and the server.

Dendro goes beyond conventional approaches (such as Semantic MediaWiki) in its adoption of LOD for representing its metadata records. Instead of relying on a relational database for storing and querying the records in the business logic, it uses a graph representation and SPARQL at its core. The first advantage of this approach is that the data records are always up to date (because there is no translation step between relational and triple-based models). Also, there is no need to implement and maintain the code that performs that periodic update of the Triple Store from the contents of the relational database that is used as the transactional system. As disadvantages, we list the slower performance of some graph databases, as well as their lack of support for ACID transactions.

Unlike most repository platforms, which are supported by relational databases, Dendro's data model is a graph and is represented in a triple store, with different ontologies establishing the semantics of its entities. All the queries performed over the database are written in SPARQL and Ontologies represented as OWL files can be directly loaded into the model. While arguably less efficient than an approach based on a relational database, this has many advantages over the latter in a data publishing context:

1. **Full Linked Data compliance**
   Every record is ready to be made available on the web as RDF, OWL, JavaScript Object Notation (JSON) or eXtensible Markup Language (XML), since it is created. The datasets can also be queried via SPARQL endpoint provided by Virtuoso (the same database technology that supports DBpedia, the so-called *nucleus for the web of Open Data*[2]) [17], allowing external systems to retrieve dataset metadata easily.

2. **Domain-specific metadata**
   The graph makes it easy to add domain-specific metadata with unique semantics. Every dataset is represented as as node with as many edges as necessary, each with distinct meanings and purposes. As an example, one can have edges to express the relationships between files and folders in the project structure, and edges to represent the metadata associated to each node, all in the same structure.

3. **Simple data model**
   Since all the data in the system is represented as a graph where every node and edge has a meaning specified in ontologies, the model

2 http://dbpedia.org

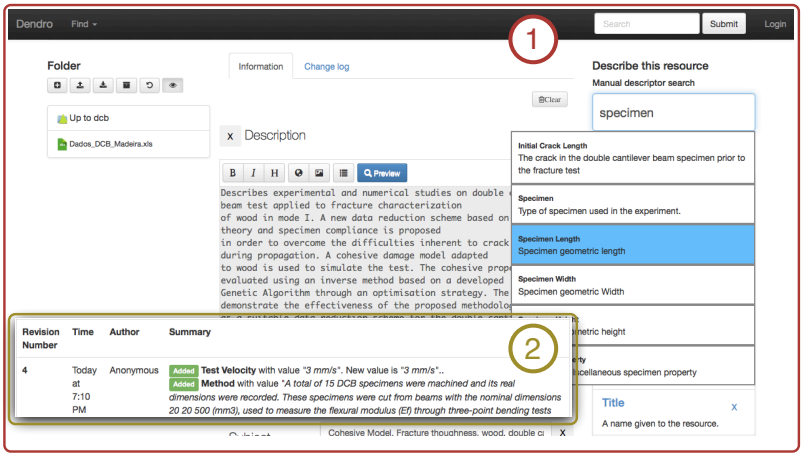


Figure 10: Dendro's data description interface (1) and change history panel (2)

becomes easier to understand when compared to a relational counterpart. This makes it easier to perform migrations (upon the inevitable decommissioning of Dendro, if we consider the long term) and makes it easy for programmers to expand the model and the platform itself (easy maintenance).

4. **Easy integration**
   Dendro has been successfully integrated with existing repository platforms that provide an open Application Programming Interface (API), such as CKAN, DSpace, Zenodo, ePrints, and Figshare. The platform's plugin-based architecture allows programmers to easily integrate data deposits into other platforms, depending on the features provided by their API.

5. **Separation between metadata modeling and programming**
   The ability of the solution to directly load OWL or RDF-represented ontologies into its data model allows curators to grow the platform's data model without need for changes in the code. With a system built on a relational database, changes to the data model would very likely mean changes in the code as well.

7.2.2 Easy to use interface

Figure 10 shows the resource description interface in Dendro. On the left hand side there is a file explorer that allows users to navigate through the files and folders in a research project. A project is a directory structure stored in the platform, much like a *Dropbox* folder, shared among the members of the research team. At the center are the descriptions already added to the currently selected folder. On the right there is an input list that enables users to retrieve descriptors to add to the current description. As users type, they are presented with a set of descriptors (ontology properties) whose label or comment annotation properties match the value entered in the box. In this particular example the selected descriptor, Specimen Length or dcb:specimenLength originates from an ontology designed specifically for the Double Cantilever Beam (DCB) domain (the DCB ontology, denoted by the dcb: namespace). Other descriptors such as dcterms:description, dc-

terms:title or dcterms:creator come from another ontology (Dublin Core Terms)[3]. More descriptors from additional ontologies can also be added. These metadata can vary greatly from resource to resource and are unknown at the time of modeling because the system has to be suitable for different domains. Since this is a collaborative system, all records must be versioned in order to keep track of past values, as well as their authors and timestamps. This is where Dendro's graph-based data model can help maintain queries simple while keeping the model itself simple to understand.

7.3 DENDRO IN THE RESEARCH DATA MANAGEMENT WORKFLOW

Dendro, as a research data management platform, aims to establish a trade-off between close proximity to the researcher, incremental data description, quick and simple deposit and no strict metadata requirements. It uses a triple store to support an ontology-based data model in order to satisfy the metadata needs of different research communities. No metadata requirements exist at the time of deposit, but the basic descriptors (creator, modification date, creation date, etc.) are provided, and the user is expected to fill them in. Richer descriptors are presented as *recommendations* that researchers and curators can choose to fill in or not for each resource. From a preservation standpoint, it is completely supported by open-source software built for horizontal scalability. Its underlying data model makes data easier to preserve due to its intrinsic readability and LOD foundation. It does not require a relational database and is designed to foster dataset integration in the Semantic Web as LOD. An interesting side-effect that stems from the adoption of this model is that the usual layers of relational-LOD translation logic that often exist in solutions that provide LOD compatibility solutions are eliminated. A practical example of the latter is Semantic MediaWiki, that uses a relational database in its transactional system and an RDF store for semantic querying, requiring specific code to maintain a permanent mapping between the two solutions. We argue that, by removing the dependency on a relational database altogether, we can remove the concerns over its migration when the system is rendered obsolete and provide an ontology-based metadata model from end to end.

The goal of Dendro in a data management environment is to act as a *staging platform* that researchers can use in their daily activities. Instead of focusing on providing sophisticated tools to researchers from a single research domain, however, the platform aims to support collaboration between researchers in different domains. By capturing datasets from the moment that they are produced, the quality of the metadata records that are produced tends to be higher because more contextual information can be gathered. As time passes the metadata is harder to gather or is not even produced at all, due to changes in the composition of the research teams, or to the closing of research projects.

The platform was designed with several goals that revolve around several desirable traits in a preservation solution.

Figure 11 shows the role of Dendro in the research data management ecosystem as it supports the process at different points in time.

3 An OWL version of the DCTERMS schema is available at http://bloody-byte.net/rdf/dc_owl2dl/index.html

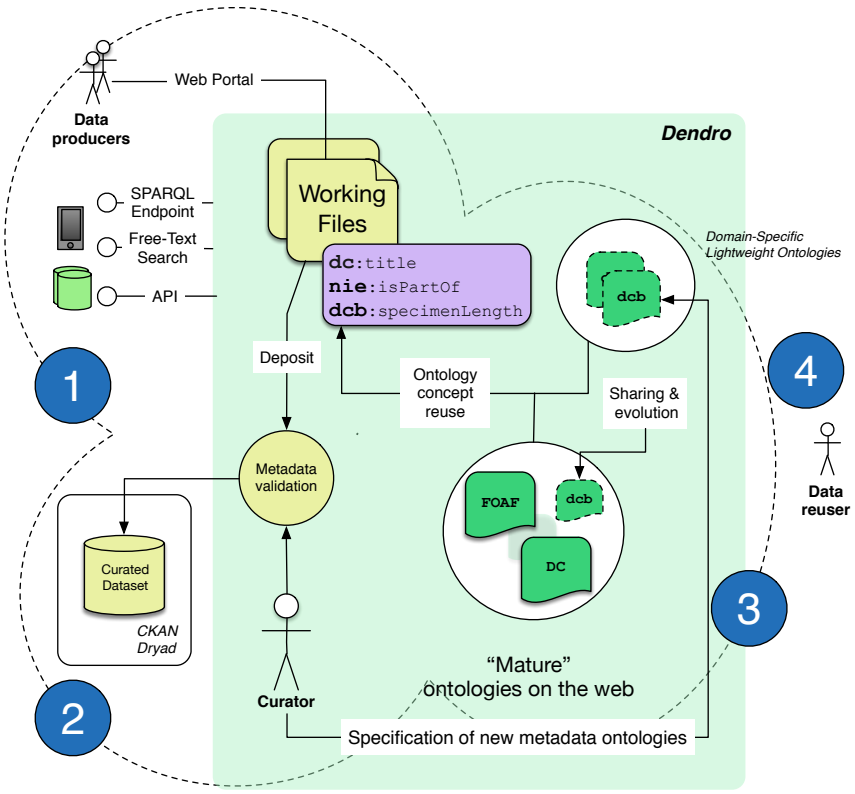


Figure 11: The role of Dendro in a research data management ecosystem

1. Data creation, description and sharing within the research group throughout their research activities (**1**). Dendro provides a friendly web interface for humans as well as a series of APIs to enable other systems to manipulate files and folders as well as their metadata. Metadata creation is carried out using properties from different ontologies (either already present on the web or modeled by curators). With a triple store as the storage and querying layer, metadata can be added as property instances. Resources can also be retrieved using SPARQL queries, making faceted searches much easier to implement than on a relational model. Moreover, the simple triple store model enables external entities to easily query the data store via SPARQL.

2. Dataset deposit, where a set of files from Dendro, as well as their relevant metadata, are packaged and deposited in a long-term preservation platform such as Zenodo or CKAN (**2**)

3. Evolution of metadata model (**3**). As the metadata specifications for different domains are created, they are also shared on the web, encouraging reuse and community-driven maintenance. Descriptor semantics become publicly documented and available for reuse in other data management systems, enabling a continuous evolution process that contributes towards the emergence of some ontologies as metadata standards for different research domains.

4. Data reuse (**4**). When a researcher accesses a dataset, documentation on the meaning of each descriptor will be available in the ontology

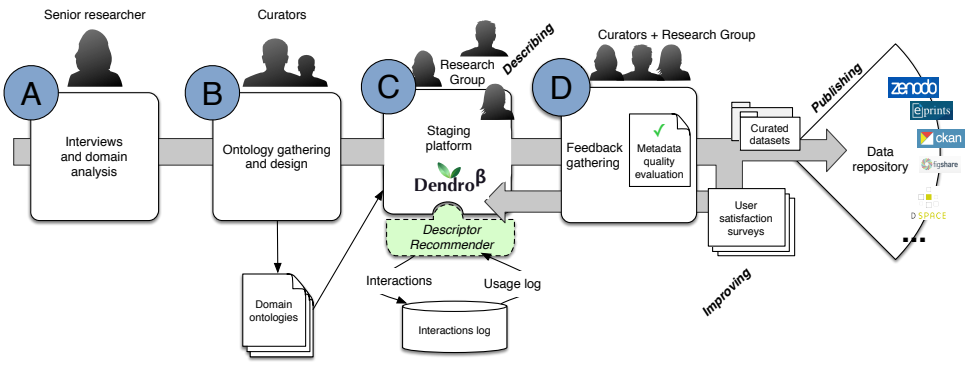


Figure 12: Dendro in the research workflow

from where that descriptor originated, making the interpretation of domain-specific metadata easier.

Figure 12 shows our proposed workflow for a research group that uses Dendro as its data staging platform. At the start of the workflow, a senior element of the group meets with an element of the research institution's curatorial team (**A**), who builds (if necessary) an ontology containing those very domain-specific descriptors that are not yet present in the platform. Other ontologies already existent on the web can also be loaded if they contain relevant descriptors (**B**). After these are loaded in the platform, the research group can start uploading and describing their datasets autonomously (**C**).

The description needs of different research groups are distinct from each other due to their domains, and there can be hundreds or thousands of descriptors to choose from; to deal with this, a recommender system is used by Dendro to automatically build Application Profile (APs) that adapt to the needs of every research group. The system records the interactions performed by the users over the descriptors available in the system; those descriptors that researchers actually use are selected among the universe of descriptors available and, in time, a domain-specific AP can emerge. In contrast with current techniques for metadata schema and full-fledged Dublin Core Application Profile (DCAP) creation [60], there are less data management knowledge requirements placed on those users who just want to record metadata with some detail but do wish to avoid the complexities of building a DCAP. In this proposed usage-based approach, the descriptors to be included in an AP emerge from their actual usage.

This approach does not intend to fully replace the existing techniques for designing APs, but rather provide valuable insight that can lead to an AP specification for an entire domain, if the investment in its formalisation is deemed worthwhile by a large enough community.

In the final step (**D**), researchers can share a subset of their projects to any of the major repository platforms, which then support all the necessary features for data publishing (e.g. metadata validation, embargoes or dissemination). The curator and the researchers work together to assess the quality of the metadata records, correct them if needed and then publish the finished dataset. Periodically, research groups are inquired about their satisfaction with the workflow and with the platform, and the gathered suggestions would serve to steer further development on the Dendro platform.

7.4 THE NEED FOR A GRAPH-BASED MODEL

In this section we perform a critical analysis of several widely used open-source data management platforms, as well as a popular semantic wiki solution. Although not specifically targeted at data management, bibliographic record management platforms tend to be adapted by institutions for dataset publishing, so we analyse a few as well.

This analysis focuses primarily on their relational models, in particular their limits regarding metadata representation. We then present Dendro, a collaborative, ontology-based platform for research data management built on a triple store. Its data model learns from ontology-based approaches [134, 133] and it is innovative in the sense that it allows domain-specific ontologies to be directly used in resource description.

A collaborative environment for research data management is conceptually located somewhere between a semantic wiki and a repository platform. Semantic wikis are oriented towards collaborative description and interlinking of resources, while research data management platforms are mostly oriented towards the end of the research lifecycle. The end result is that their relational models are designed around different concepts to cope with different requirements. Good examples are the time dimension in systems where versioning is very important, or the metadata dimension in those that need to support several metadata schemas for their datasets.

In any wiki (including semantic wikis), time is a crucial dimension. Pages content must be versioned and must be easy to query according to their date of creation. On the other hand, most research data management platforms consider datasets as static resources with constant attributes; this happens because these solutions are not designed to accompany the activities of research groups, relying on a late timing for deposit—typically only after the results that derive from those datasets are published. As a consequence, current data repositories do not handle the versioning of the deposited files or their respective metadata, as the submitted version will already be a final one (CKAN is a notable exception to this behaviour).

For our analysis, we selected only open-source platforms, since closed-source alternatives are effectively a liability in workflows oriented towards long-term preservation. By promoting vendor lock-in, they constitute a single point of failure; should the company developing the data management software leave the market, all the data stored in such platforms will become at risk [57]. Dealing with open-source software, we were able to explore their relational schemas on our own installations in order to draw conclusions based on hands-on experience.

7.4.1 DSpace

DSpace is a widely used open-source repository platform designed for managing academic publications. It is not targeted specifically at research data management but, being an open-source project, it can be adapted to become a more data-oriented solution. It has been adapted in the past to support basic data-oriented features like in-browser exploration and querying, as well as dedicated dataset search features [192, 193]. Among open-source repository platforms, DSpace stands out because of its large number of running installations around the world. The platform's storage and querying layer adopts the following main concepts in its relational database:

1. Item—a set of files, or Bitstreams, to which metadata descriptors, typically Dublin Core, can be associated;
2. Bitstream—a data file within an Item. These can be images, spreadsheets, documents, etc. Bitstreams have no individual metadata apart from internal control fields;
3. Metadata Schemas—groups of Metadata Descriptors that have a namespace (for example, dc for the Dublin Core Terms);
4. Metadata Descriptors—each of the descriptors that can be instantiated for an Item in order to record their metadata.

An excerpt of the relational model for DSpace 3 can be seen in Figure 13, area **1**. For each Item and Metadata Field, there can be several values, recorded in the metadatavalue table. As can be seen from the structure of the metadatavalue table, the values are internally saved as text values.

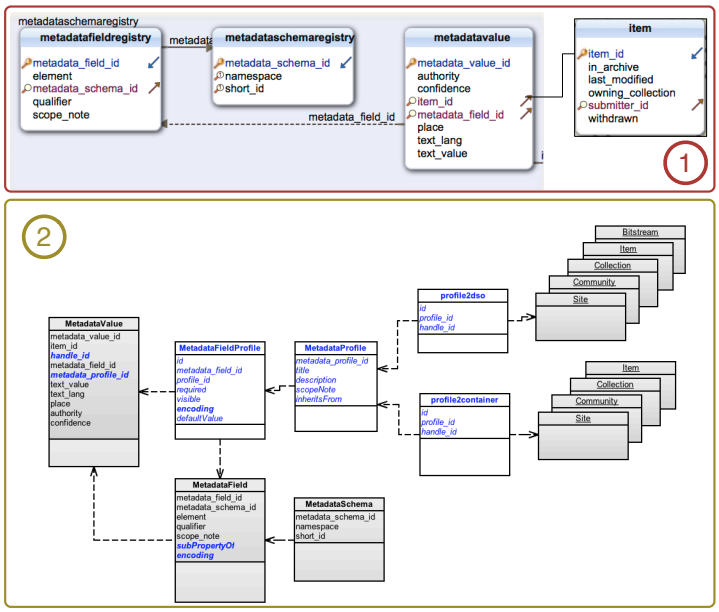


Figure 13: Relational model of DSpace 3.x (1) and improvement proposal (2) (Image source: DSpace wiki)

This metadata model is rather simple to understand. It is easily parametrizable, supports multilingual metadata values, and allows descriptors to be grouped according to the schemas to which they belong. On the other hand, there is no support for different types of metadata values (for example int, string or date), at least in an explicit way, and they are not versioned.

A DSpace proposal[4] for an updated model for metadata recording in DSpace can be seen in Figure 13, area **2**. This proposal aims at providing a more extensible data model for representing metadata added to DSpace objects, addressing several concerns:

1. Defining metadata *profiles*, i.e. sets of descriptors that can be attributed to the different types of DSpace objects, since the current model only provides support for Item metadata;

4 Description available at https://wiki.duraspace.org/display/DSPACE/Proposal+For+Metadata+Enhancement

2. Providing a descriptor hierarchy, much like properties and subproperties in an ontology; however, a relational approach requires the implementation of *inference-like* logic;

3. Using certain *established* schemas, for example DC, as extension points from which others can be derived (much like sharing ontologies on the web for others to build upon);

4. The ability to validate metadata records based on different schemas—to check, for example, if all the descriptors needed for compliance with a metadata schema are present in a metadata record (this can be achieved by validating class membership in an ontology-based solution);

5. Restricting the metadata that can be assigned to a resource depending on the type of that resource (in ontologies, this is achieved simply by specifying the domain of a property).

While this model would effectively satisfy the elicited requirements, it would add another layer of complexity to the relational schema, incurring in heavy JOINs. At the same time, it does not cope with the need of keeping track of past versions of metadata records, something that we argue is an essential requirement for a collaborative data management system. An ontology-based model, on the other hand, can address these concerns, provide support for time-dependent metadata values and keep the model simple to understand. When implemented using a triple store, such a model requires less knowledge of the particular database schema and enables *closer to domain semantics* querying via the SPARQL querying language.

7.4.2 Zenodo (Invenio)

Zenodo[5] is a data repository service designed for the deposit and publication of research datasets. According to its documentation, Zenodo is a layer built on top of Invenio[6], a repository platform for bibliographic assets such as papers, thesis and preprints.

Figure 14 presents some aspects of the Invenio platform. Area **1** shows a section of the dataset deposit interface, which includes a set or pre-defined metadata fields that the user must fill in. Area **2** shows an excerpt of the tables present in the schema.

At the time of installation, the schema included 541 tables, many of them with very similar names. No foreign keys are specified in this relational schema. Given the complexity of this schema, we were unable to determine precisely how it records metadata values, even after analysing schema backup scripts; it is however apparent that there are strong ties between the relational model and the business logic code in the platform. The lack of foreign keys and the obscure table names make it very hard for potential contributors to understand where information is stored, so our opinion is that this is not a good alternative at the present time. The complexity of its relational model will be an obstacle to maintenance, while making it hard to migrate the contents of an Invenio database when the platform becomes obsolete.

5 http://zenodo.org
6 https://github.com/zenodo/invenio-op. Original Invenio page: https://invenio-software.org

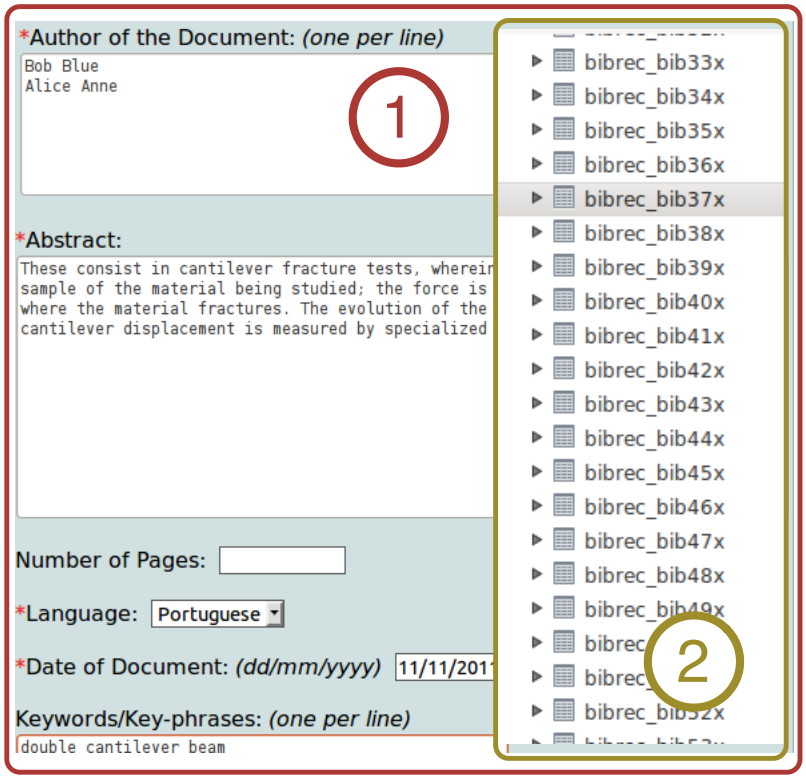


Figure 14: Invenio's deposit interface (1) and a a partial list of database tables (2)

7.4.3 CKAN

The CKAN is an open-source data publishing solution supported by the Open Knowledge Foundation (OKF) designed to be easily integrated into and extended by other services. One of the strong points of the solution is that it provides a core set of features and then allows individual instances to be customized according to the instance maintainer's specific requirements. This interoperability is necessary to provide federated search, where data assets from many CKAN instances can be retrieved through a single instance. The OKFN is a driver for the adoption of open data policies in UK institutions and CKAN a tried and tested solution[7].

CKAN requires minimal metadata throughout the dataset deposit workflow. While some may argue that this may yield substandard descriptions with low metadata quality, it has been also argued that this is the right choice [101, 204]. We argue that metadata production should be reduced to a minimum in the first stages of the process to encourage researchers to make a first data deposit. Richer metadata may be added as necessary, not only as a way to record and share information within the research group but also to increase the chances of data and publication citation due to easier dataset discovery and reuse. Ideally, this decision to add metadata should come from the researcher and not from a mandatory dataset deposit policy by the research funding institution.

Figure 15 shows the final step of the deposit process of a dataset in CKAN. As shown in the image, the metadata that can be added to the dataset in-

7 CKAN case studies include Data.Gov.Uk, PublicData.eu and the Helsinki Region Infoshare online service http://ckan.org/case-studies/

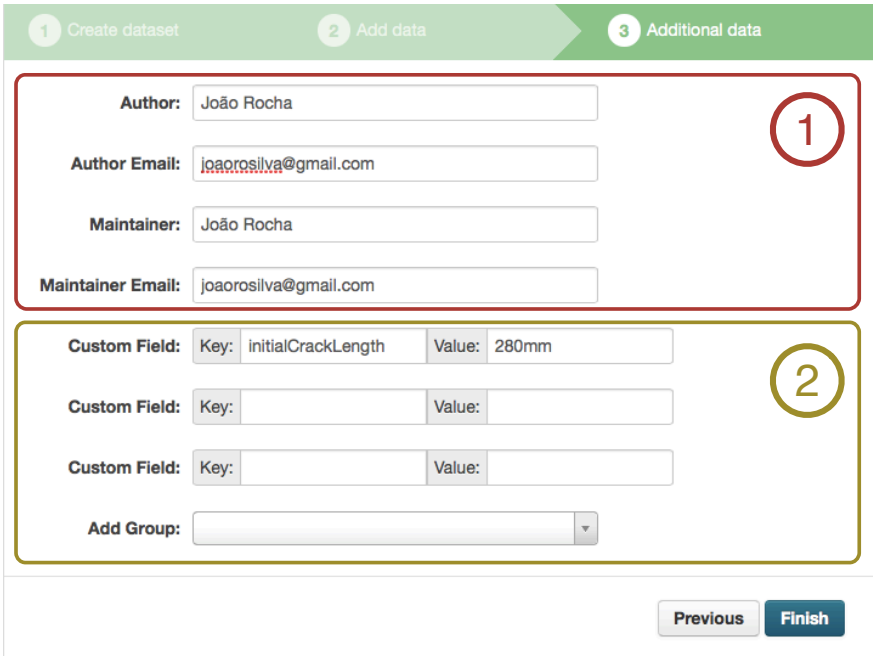


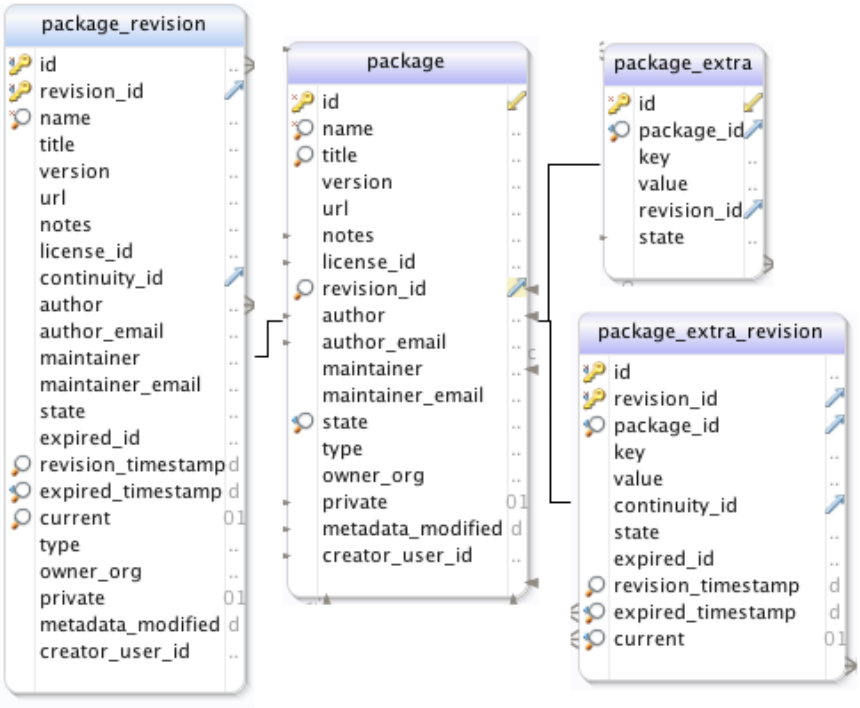
Figure 15: CKAN deposit interface

cludes a set of predefined fields (area **1**) as well as a set of arbitrary key-value metadata pairs (area **2**). These can serve to record domain-specific metadata, but the specification of adequate descriptors is left to the user depositing the data. There is no suggestion of possible descriptors nor do they originate from a controlled vocabulary or metadata schema. There are, however, several *standard* metadata fields: *author*, *maintainer*, or dataset *state* for example. Complementing these generic metadata are a set of key-value metadata records that can be used to record *domain-specific* metadata (area **1**). Despite being displayed in the same area of the web interface, the relational model actually makes a separation between the different fields, saving the *standard* metadata and the complementary *key-value* fields in different locations.

CKAN stores the metadata values in three database tables, as shown in Figure 16. The package table holds information about each submission, the package_revision holds records of the past values of the metadata relative to the submission, and the package_extra_revision table holds all the values of the key-value metadata fields specified by the user. The CKAN platform is, at this point, the solution that comes closer to our proposed approach. It places minimal barriers to data deposit by reducing mandatory metadata descriptors to a minimum while supporting time-changing metadata and versioning. Its metadata model could be improved, however, with the inclusion of descriptors from metadata schemas in place of the key-value approach for *extra* metadata.

7.4.4 Semantic MediaWiki

SWM is an extension to MediaWiki, the open-source software platform that supports Wikipedia, the world's largest online community-managed encyclopedia [220]. SWM allows users to add semantics to the links specified in




```
1 SELECT *
2 FROM
3     public.package_extra_revision per JOIN public.package p
4     ON per.package_id = p.id
```

id	package_id	key	value	revision_id	revision_ti...
b782011...	1448ff8...	initialCrackLength	280mm	ab27d64...	03.04.2...

Figure 16: Tables for metadata in CKAN's relational schema

the text of wiki pages using special markup [169]. SWM was designed to handle the requirements posed by collaborative content production (versioning, concurrent editing, modification auditing, etc.). Thus, the platform's relational model is designed around the concepts of *page* and its *revision*s which represent the different versions of that page's content. The actual contents of each page revision are stored (as a wiki syntax-formatted string) in a record within the *text* table in the database.

In SWM there are *special pages* that can be edited using *semantic forms*, specified using *templates*. While regular wiki pages are edited using a text area where users can write hypertext in SWM markup, *semantic forms* provide a predefined interface for users. This distinct interface is suited for filling in sets of metadata descriptors and hides the special syntax required for the addition of semantics to wiki links that users need to employ when using the standard text editor [98]. When the information is saved, the properties edited in the form are saved as a page revision. SWM uses ontologies to establish the semantics of the inter-page links that can be used in wiki page text. This is advantageous for preservation because ontologies can be freely shared and reused on the web, encouraging their evolution at a faster rate than metadata schemas—as the latter usually require broad agreements on any proposed change.

Listing 1: Defining the DCB ontology in SWM

```
//Defining the DCB ontology in Semantic MediaWiki

http://dendro.com/ontology/dcb/—[http://dendro.com Double Cantilever Beam]
 specimenLength—Type:Number
 specimenWidth—Type:Number
 specimenHeight—Type:Number
 specimenInitialCrackLenght—Type:Number
 specimenProperties—Type:String
 temperature—Type:String
 moisture—Type:String
 testVelocity—Type:String
```

To load ontologies into SWM, one must represent each ontology (including its properties) in SWM's own format. Loading the Friend Of A Friend Ontology (FOAF) ontology, for example, requires the creation of a special page for the new ontology. Listing 1 shows an example of how an ontology is loaded into Semantic MediaWiki. Note its ability to set the type of each descriptor (String, Number, etc). After loading the ontology, its properties can be used in the specification of semantic links within the text of the wiki pages. Describing a dataset using SWM could be achieved by the following sequence of steps:

1. Create a new wiki page with a meaningful URI for the dataset (for example http://wiki_root.com/index.php/dcb-mechanics/DCB_Madeira.xls)

2. Upload the files to the page and create a link to it in the wiki page (for download)

3. Describe the dataset by editing the current page. This can be performed by means of *semantic links* (inline markup like in Listing 2) or *semantic forms* (static templates for filling in sets of descriptors specified *a priori* during system parametrization).

Listing 2: Dataset description in SWM

```
The samples were prepared with an initial crack length of [[initialCrackLength::280]].

The width of each specimen was [[specimenWidth:20]], the height was [[specimenHeight
    :10]] and the length was [[specimenLength:200]].
```

We argue that the best alternative would be to combine the flexibility of semantic links with the structure of a semantic form to separate metadata fields and standardize description. A modified version of Semantic MediaWiki was combined with a *Dropbox*-like platform to serve as a collaborative dataset description platform [188]. In a first phase, researchers deposit their datasets (represented as files and folders) into a shared data storage area. The uploaded files and folders can then be described in the modified SWM environment, much like the filling in of a metadata sheet, but with the advantage of collaborative work and semantic description. This modified version of SWM introduced more flexible *forms*, so that users can dynamically add descriptors from different ontologies to their descriptions. This *mixing and matching* approach was an attempt at creating instances of *semantically-enabled* APs (combinations of subsets of metadata schemas designed for a particular application [94]). Without requiring the pre-configuration of the templates, researchers could fill in more or less metadata descriptors as they saw fit.

In order to provide a SPARQL endpoint, Semantic Mediawiki relies on a separate triple store to represent its semantic information. A mirroring process is required to map the contents of its relational database over to the triple store[8]. This requires maintenance operations on the code that performs the mappings, induces redundancy and is expensive in terms of processing power. We argue that a completely graph-designed solution built on a triple store and SPARQL could prove to be more adequate as it would eliminate the need for this mirroring logic to keep SWM's triple store up to date. By using a triple store, one would also dispense ontology parametrization and directly load ontologies from the web into the system.

7.5 CONCLUSIONS

Research data management platforms have recently began to appear as open-source projects supported by developer communities as well as companies. When comparing them with more mature solutions designed for managing bibliographic records (DSpace and Invenio for example), some similarities become apparent. The existence of a deposit workflow, metadata standards compliance and organization of resources into collections (or similar concepts) are some examples of common requirements.

Past studies, including our own, have determined that data management should start as early as possible. The data should be managed collaboratively among the members of the research group. Most current data management tools, however, are not designed for this scenario. CKAN takes a first step towards this: there is some support for metadata versioning, but the flexibility of its metadata model is limited to sets of key-value metadata.

When building a collaborative system, an important requirement is versioning. Implementing revision history auditing capabilities for metadata

8 As per the current documentation available at https://semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores

records when the metadata descriptors are unknown at the time of modeling always adds another layer of complexity, as can be seen in Semantic MediaWiki and CKAN, with their *revisions* tables. In a collaborative environment, however, all changes must be recorded and easily retrievable using simple queries. Wikis are designed around these requirements, and semantic wikis go further by allowing users to express relationships between the resources that they describe—in that sense they are very close to a graph.

After analyzing our requirements, and given the importance of data model simplicity in a long-term preservation context, we concluded that a relational database would not be the best solution. Its relational model would become too complex, requiring heavy and complex queries. The need for a modular, ontology-based metadata model with full versioning capabilities has led us to opt for a triple store as the transactional system.

8 DESIGN AND IMPLEMENTATION OF DENDRO

This chapter describes the technological choices behind the implementation of Dendro. From the graph-based data model implemented on a triple store to the JavaScript/JavaScript Object Notation (JSON) stack, we outline how Dendro interacts with its different support systems, and how the data model supports the engineering challenges behind building a solution of this kind.

8.1 INTRODUCTION

Dendro is the prototype research data management platform that was designed and built throughout this work. It was designed to be simple to use by users with only basic knowledge in data management. It acts as a file storage, description and sharing platform, combining a Dropbox-like interface with some features usually found in a wiki. Users can upload files and create and organise folders, while collaboratively producing metadata records for those files and folders [189]. This collaborative metadata production introduces a change in the common workflow of *a posteriori* data curation, where research assets are described after the research effort is concluded and the findings are published [190].

Dendro[1] is intended to be a staging area for research datasets. The platform targets the daily data management needs of researchers, helping them centralize the data within their research team and to describe that data collaboratively. Dendro does not intend to replace existing long-term preservation solutions, but rather to act as a first entry point for research datasets in an integrated research data management workflow, where the last step is the deposit of the data in one of these platforms.

Using Dendro, research teams can create projects to manage their data. A project can be regarded as a shared data storage area, complemented with wiki-like features designed to help researchers produce metadata records. Since researchers are not data management experts, the platform tries to balance ease of use with standards-based metadata production. It gives users the freedom to describe their data the way they want, with more or less metadata descriptors as necessary. The easy-to-use interface separates the user from the complexity of ontology-based representations, but using them "under the hood" to represent the metadata that the user produces. The result is high-quality, time and research domain-dependent metadata

1 Video demonstrations for Dendro are available; short version (approx. 4min): http://goo.gl/ug4FTh. Long version (approx. 40min): http://goo.gl/SvdXhd.

that can be added to file and folder hierarchies [197]. Targeted at researchers and curators alike, Dendro aims to support users with basic data management background in their basic data description, offloading data curators from metadata creation tasks. The role of the curator does not completely disappear, however—curators still play a vital role in validating the records produced as the data is being created. The goal is to collaboratively improve those records from the time of data creation, in order to ensure that the datasets and respective metadata records can be later deposited in a data repository, thus ensuring their long-term availability.

The platform has an extensible data model built on ontologies, which allows users to easily "mix-and-match" descriptors from different metadata schemas in their metadata records, similarly to the building of an Application Profile (AP) [94]. The simplicity of its data model also facilitates the platform's decommissioning when it becomes necessary to do so, increasing the survivability of the data and metadata contained inside. When compared to existing solutions supported on a relational model, its triple-based data model combined with full Linked Open Data (LOD) compliance allows the representation of domain-specific metadata in a more flexible, interoperable and (arguably), simpler way when compared to other repositories [197]. These often implement protocols such as Open Access Initiative—Protocol for Metadata Harvesting (OAI-PMH) or specific Representational State Transfer (REST) Application Programming Interfaces (APIs) to expose their metadata records to external systems [13], instead of relying on Uniform Resource Identifier (URI) dereferentiation, as proposed by the Linked Data guidelines [31].

Dendro is not the only solution that implements collaborative data management workflow. The ADMIRAL project has adopted a similar approach, with a combined implementation of DataFlow, a solution that provides robust shared storage space for researcher groups to share their data, and DataBank, a secure repository for long-term dataset deposit [101]. EUDAT, a pan-European collaborative data infrastructure also recently presented B2Share, a modified version of the Invenio research data management software platform[2], and its roadmap includes the integration into a full long-term preservation workflow [91].

This chapter is based on the following publications [190, 189, 10]:

- Rocha, J., Castro, J., Ribeiro, C., & Correia Lopes, J. (2014). The Dendro research data management platform: Applying ontologies to long-term preservation in a collaborative environment. In Proceedings of the iPres 2014 Conference

- Rocha, J., Castro, J., Ribeiro, C., & Correia Lopes, J. (2014). Dendro: collaborative research data management built on linked open data. Proceedings of the 11th European Semantic Web Conference.

- Amorim, R., Castro, J., Rocha, J., & Ribeiro, C. (2014). LabTablet: semantic metadata collection on a multi-domain laboratory notebook. In Proceedings of the 8th Metadata and Semantics Research Conference (MTSR 2014). Springer

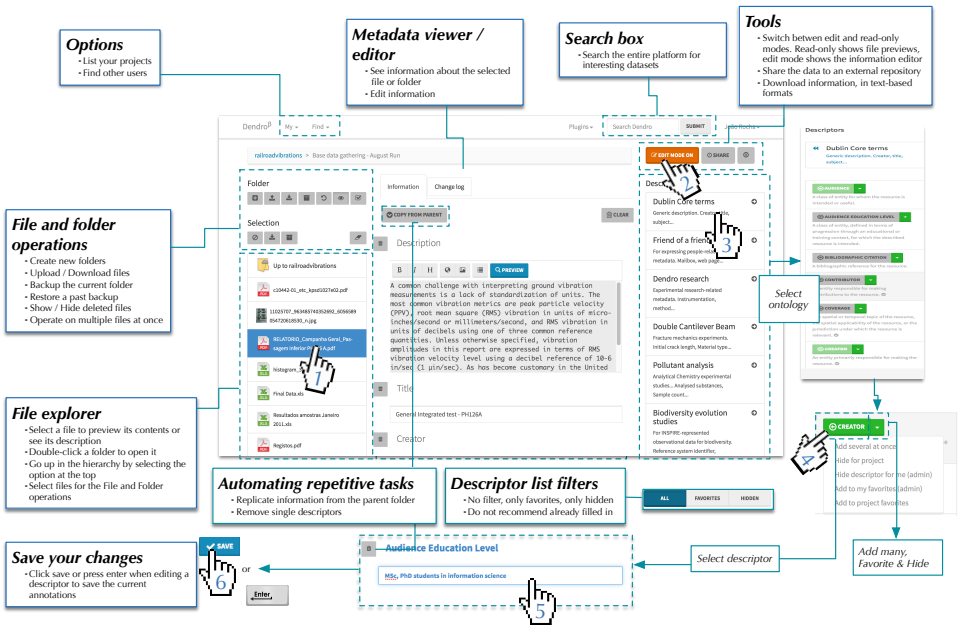


Figure 17: Using Dendro to describe research data

8.2 THE STANDARD DENDRO INTERFACE

Figure 17 shows the interface of Dendro in a standard form, without any descriptor recommendation features enabled. The mouse pointers shown in the figure illustrate a common sequence of actions in the description process:

1. Select the file or folder to be described
2. Switch to edit mode to enable the description features
3. Select an ontology from which to select descriptors to be filled in
4. Select the desired descriptors (will be added to the metadata viewer/editor)
5. Fill in the descriptors
6. Save the changes to the metadata record.

On the left there is the *File explorer* that allows users to navigate through their projects' file and folder hierarchy, with several buttons at the top. The first three buttons allow the user to perform common operations, such as: to create a new folder inside the current folder, to upload files, to download the whole folder as a ZIP file and to to back-up the folder (same as the download folder operation, but including also the all metadata in TXT and Resource Description Framework (RDF) format). The last three buttons allow the user to restore the current folder to the contents of a previous backup, to show or hide deleted files[3], and lastly to turn on a multi-operation feature, which allows users to select multiple files and perform operations over

2 http://invenio-software.org/

3 When a file or folder is "deleted" for the first time, it is actually hidden, and only really deleted after a second delete operation, emulating the behaviors observed in popular cloud storage solutions such as Dropbox

them (multiple downloads or multiple deletes, for example). The metadata viewer/editor occupies the center portion of the interface.

The interface works in two modes: editing mode and read-only mode. If the interface is in editing mode—"Edit Mode On", as highlighted by No.2 pointer—the descriptors at the centre show as editable text boxes, date picking fields or map previews, depending on the data type of the descriptor. At the top, the user has access to a toolbar with options to save the changes made, "Copy from parent" (copies any descriptors from the parent folder), and the "Clear metadata" option which deletes all descriptors associated to the current folder or file. The descriptor inheritance feature was added as a time saver to facilitate description tasks of those elements of research groups that perform repetitive data gathering tasks, and which often yield datasets with similar metadata. Using this feature, they can describe a *parent* folder with the descriptors that are common to all experiments and then copy the metadata to all child files or folders contained in that parent folder. Descriptors can also be individually removed from the metadata record by pressing the "garbage can" buttons next to them, and saving the changes.

If the interface is in read-only mode—"Edit Mode Off"—the editor at the centre will change into a view more suited for reading, with all the descriptors organized in a compact manner, and a preview of the selected file at the top. For Portable Document Format (PDF) files and images, for example, a visual preview will appear; for Excel files, a grid-like preview will be shown; for multimedia files, a play/pause control and a seeker control will be displayed.

When the interface is in editing mode, the right-hand side displays the list of ontologies from which users can select the descriptors to fill in. After selecting an ontology, the user will see the list of descriptors contained in it, ordered alphabetically from those starting with *A* (at the top of the list) to those that start with *Z* (at the bottom of the list). After selecting the descriptors and filling them in, the user must save the changes.

Next to each of the descriptor selection buttons there is a small button with a downward pointing arrow, which provides several tools related to the descriptor. It allows users to add multiple instances of the descriptor to the metadata editor (makes it easy to add multiple Contributors or Subjects, for example), to set interesting descriptors as "Project favorite" or "User favorite" or to hide uninteresting descriptors, either for the user or for the project. Hiding a descriptor for the user will cause the descriptor not to be presented to the user anymore. Hiding a descriptor for the project will prevent the descriptor to be shown for all contributors of that project, but only when they are navigating that project's file and folder structure.

8.3 SHARING AND EXPORTING FUNCTIONALITIES

Since it is designed to serve primarily as a staging platform, Dendro allows users to export data and metadata in a multitude of ways. Figure 18 highlights the possibilities available to users for publishing their data and metadata. The **A** part highlights a part of the interface shown in Figure 17, which allows the user to export the metadata of the currently open file or folder in text format, RDF or JSON. In the case of a folder, it will export the metadata about all its subfolders and files that it contains as well. The text representation is formatted to be easily readable (for humans), while the other two formats are good for automated processing by machines. To

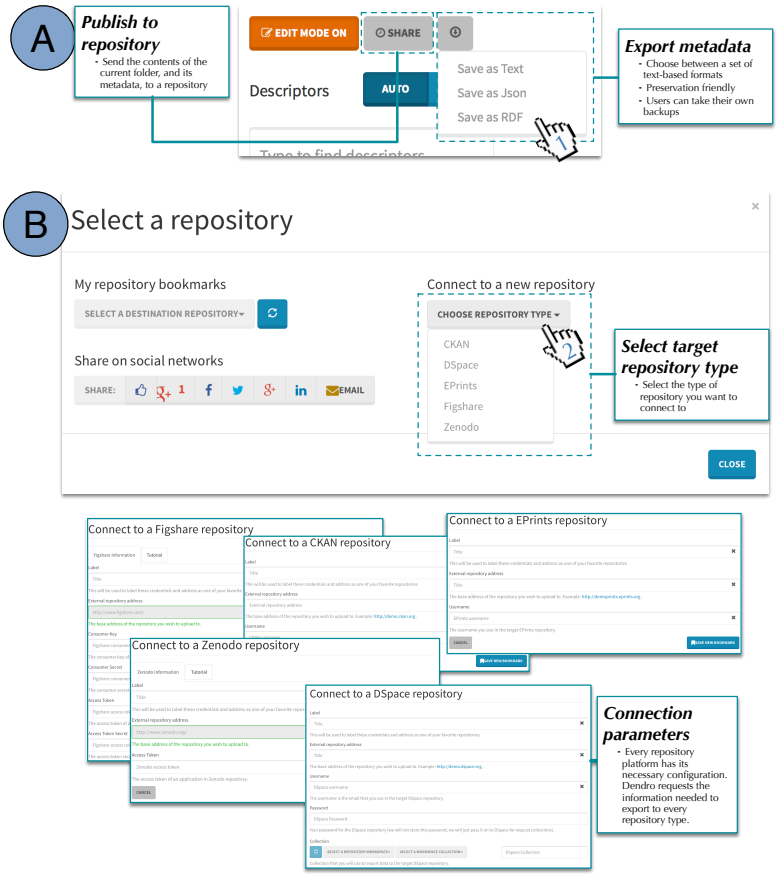


Figure 18: Data exporting and publication capabilities in Dendro

the left, there is a "Share" button that allows users to publish the contents of the currently open folder to a repository of their choosing. After pressing the share button, they are presented with a popup (Area **B** of Figure 18) where they can select the type of repository to which they want to share and enter a series of connection details that should be provided by the target repository's IT administrators. The values need only to be input once, after which the connection details will be saved as a bookmark for later deposits. Information such as the address of the repository, access tokens and other API-specific details are kept; sensitive information such as usernames and passwords for the external repository are only used for that specific deposit operation and are not stored in plain text within Dendro (for security reasons).

8.4 THE DATA MODEL OF DENDRO

A flexible data model is necessary to cope with the requirements of the various research domains. Researchers from different domains should be able to build and use their own "APs"—sets of descriptors tailored to the needs of the research group, albeit without the same degree of formalism as what is present in a Dublin Core Application Profile (DCAP).

In a relational model, the different system entities are usually mapped to separate database tables, and the relationships that exist between them mapped as foreign keys. Such a representation shows its limitations in some particular cases, however—three specific requirements were identified as essential for the implementation of Dendro's use cases.

- **Entities with distinct attributes**
  When a system has to represent entities whose attributes are not entirely known at the time of modelling, their underlying relational database schema can quickly become convoluted. The most obvious way to cope with this is to represent attributes and entities as records in distinct tables; the values of each attribute-entity pair (if it is present) must be represented in a table with two foreign keys (one referring the described entity and another referring the attribute) as well as the respective value of the attribute for that entity.

  Existing systems such as Comprehensive Knowledge Archive Network (CKAN) follow this approach, but these values are limited to string values. The reason behind this is that relational systems are "strongly-typed", requiring the specification of a single type for each database column (integer, date, text...). The text type is selected because it allows the database to store numerals as well as text, even if those numerals have to be re-parsed later if to recover their numeric form. It is easy to see that there will also have to be some metadata about those values, namely for recording if the textual value represented in the database table represents a numeric type, and if it is the case, which type (float or integer, for example) should it be parsed into—this greatly complicates database queries and may require the use of views, with obvious impacts on efficiency.

- **Hierarchies between resources**
  Relational database systems often struggle to represent hierarchies between resources. The most common approach followed in such systems is to add a column to a table representing resources in a hierarchy,

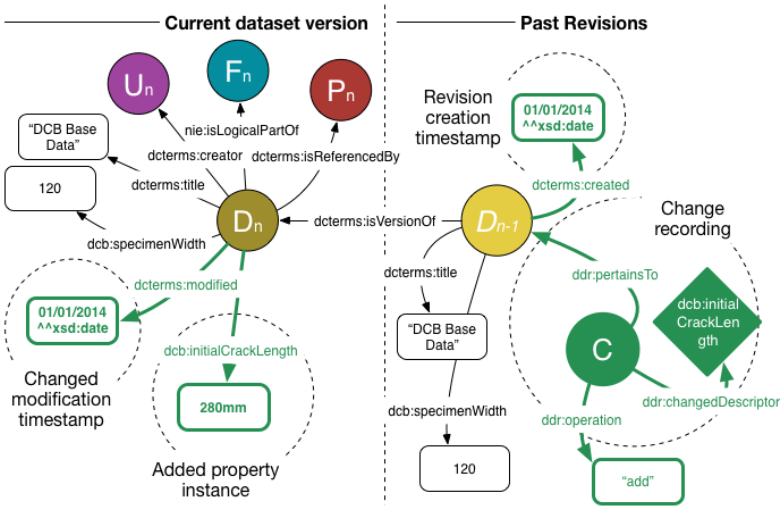


Figure 19: Dendro's graph-based data model

containing a foreign key that targets the primary key of that same table; the file-folder hierarchy is a classic example. This approach shows its limitations, however, when the hierarchy is established between resources of different types, which can be unknown at the time of system modelling. Also, navigating up the hierarchy to look for a given resource or resource type requires the programmer to write several stored procedures to cope with each case. The use of a triple-based model with SPARQL Protocol and RDF Query Language (SPARQL) support, on the other hand, allows simple retrieval of such resources because of its ability to perform *inference* over the information that is represented in the database. In this case, the *transitive* nature of a *isPartOf* property allows the system to navigate recursively up the hierarchy until the desired resource is retrieved.

- **Change history of metadata records**
  In a real-world research data management scenario, research group members need to keep track of the changes made to the files and folders that contain their datasets.

Figure 19 illustrates Dendro's data model. In Dendro, all resources can be versioned; on the "Current Dataset version" all resources have an n subscript, meaning that the resource is the latest version of the corresponding entity. On the Past Revisions section, the most recent past version of file D is shown, identified by the $n-1$ subscript.

In the situation depicted in Figure 19, a researcher uses Dendro to record the *initial crack length* (domain-specific metadata) of the samples used in the production of D, a dataset for a Double Cantilever Beam (DCB) experiment. An instance of dcb:initialCrackLength, a property from the DCB ontology, is therefore added to D's metadata values. The changes between D_n (the new version of D) and D_{n-1} are highlighted in dashed circles. First, a copy of the current resource is created. It will be a new resource identified by its own URI, and will be an instance of ddr:ArchivedResource, a class from Dendro's own ontology. It will have its dcterms:modified property replaced with a dcterms:created instance (current date) to keep track of the versioning history. All the other current properties of D will be recorded in D_{n-1}. The

Table 4: Categories of resources in the Dendro graph model in Figure 19

Element	Category	Source ontology	Class
U_n	*Resource*	Dendro	ddr:User
F_n	*Folder*	Nepomuk Information Element	ddr:Folder
P_n	*Paper*	Dendro	ddr:Paper
D_n	*File*	Nepomuk Information Element	nie:InformationElement
D_{n-1}	*File*, *Archived Resource*	Nepomuk Information Element, Dendro	nie:InformationElement ddr:ArchivedResource
C	*Change*	Dendro	ddr:Change

new metadata value is recorded as a instance of the dcb:initialCrackLength property, with D as its subject. The date of last modification (instance of dcterms:modified) is set to the current date.

To record a detailed account on the changes made to a resource, a series of Changes will be created for each descriptor that is added, removed or changed. This allows a fine-grained analysis of the changes performed for audit and usage analysis purposes. The instance C records the edited descriptor (in this case, dcb:initialCrackLength) and the type of change ("add" for descriptor addition).

Table 4 lists the different categories of resources in Figure 19, their corresponding ontology classes, as well as the source ontologies where these classes are drawn from. For example, the F_n vertex shown in Figure 19 is a *Folder* according to Table 4, a concept sourced from the Nepomuk File Ontology. In addition to these, more classes and properties from other ontologies can be included in the data model, thus contributing to its flexibility.

8.4.1 Implementation and querying

In Dendro all queries are written in SPARQL, which is executed on an instance of OpenLink Virtuoso Open Source[4]. Some example queries are provided to illustrate the simplicity and expressiveness of this data model in comparison with a relational approach. Listing 3 shows a query that retrieves all the versions of a resource, as well as their metadata values. The flexibility of SPARQL makes it possible to retrieve metadata values with different types (integers, dates, strings and resources of different classes) transparently. In a graph model there are no implicit semantics making it easier for a programmer from an external system to query the knowledge base—this makes SPARQL queries more readable since previous knowledge of the database schema is not required.

4 See http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/

Listing 3: Getting all the versions of a resource

```
SELECT *
FROM <http://127.0.0.1:3000/dendro_graph>
WHERE
{
        ?version ddr:isVersionOf <http://127.0.0.1:3000/project/dcb/data/Base%20
            Data>.
}
```

Another query example is shown in Listing 4. This query fetches all the changes associated to the latest revision of a resource.

Listing 4: Getting the changes of the latest versions

```
SELECT *
FROM <http://127.0.0.1:3000/dendro_graph>
WHERE
{
        ?version ddr:isVersionOf <http://127.0.0.1:3000/project/dcb/data/Base%20
            Data>.

        {
                SELECT ?latest_version_nmbr
                FROM <http://127.0.0.1:3000/dendro_graph>
                WHERE
                {
                        ?version ddr:versionnmbr ?latest_version_nmbr.
                }
                ORDER BY DESC(?latest_version_nmbr)
                LIMIT 1
        }

        ?version ddr:versionnmbr ?latest_version_nmbr.
        ?change ddr:pertainsTo ?version.
        ?change ?p ?o.
}
```

8.5 TECHNOLOGICAL ANALYSIS

When designing the tools for an integrated preservation environment, one must ensure that the data stored within can survive the obsolescence of the environment itself. Dendro's triple-based data model, its reliance on shareable ontologies and a full open-source technology stack all contribute to maintaining access and interpretation of the stored datasets even after the platform's decommissioning.

Figure 20 shows the architecture of Dendro. The "Data" layer holds the data model for the platform, composed of three subsystems: an OpenLink Virtuoso Database (Open-Source version), an ElasticSearch server to enable distributed document indexing and a MongoDB/GridFS file storage cluster. The graph database is used to represent all the resources in the knowledge base, such as Researchers, Files, Folders and their attributes, represented using existing ontologies. Some of the ontologies being used at this time are Dublin Core Terms Ontology (for all resources in general), the Nepomuk File Ontology (for files and folder structures representation) and the Friend

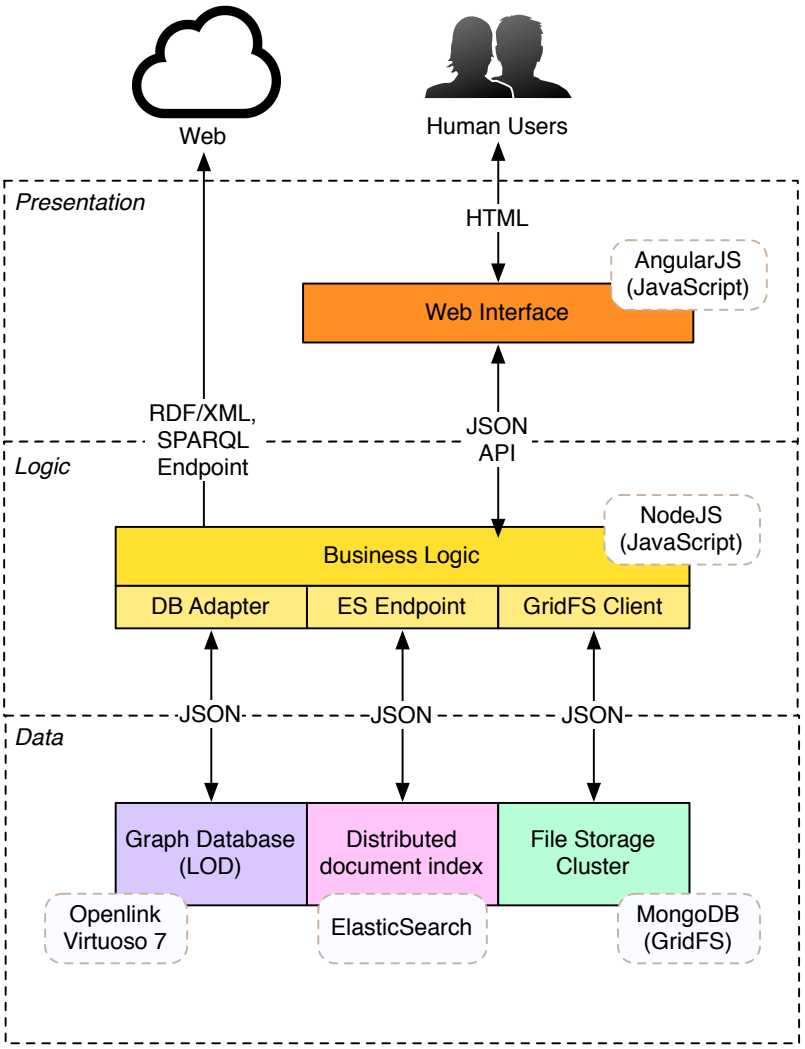


Figure 20: Dendro's architecture and technology stack

of a Friend Ontology (for describing platform Users). All queries specified by the *Logic* layer are sent to OpenLink Virtuoso's SPARQL endpoint. In case Virtuoso becomes obsolete, Dendro's triple-based model is designed to live on, since it can be fully exported in RDF and imported into another RDF-compliant solution. The triples plus the ontologies made available on the web enable a robust understanding of the stored information.

The *Logic* layer comprises Dendro's business logic, and includes three endpoints that connect to the underlying Data layer. A Database Adapter was written from scratch as part of this work, in order to provide a higher level of abstraction over the REST API provided by OpenLink Virtuoso. The module automatically performs the conversion between the results format provided by Virtuoso and Javascript objects to provide programmers an abstraction over the database, Object-Relational Mapping (ORM), similar to Hibernate for Java or the Language Integrated Query (LINQ) in the .NET platform.

The Logic layer is written in NodeJS for handling large numbers of simultaneous connections—this allows numerous users or external systems (via the Dendro API) to interact directly with the platform to manage data and metadata. Dendro is primarily written in JavaScript, a simple and very

widely known and used programming language among web developers—a plus when planning for an open-source preservation effort, as a large potential developer base makes maintenance and evolution easier.

8.5.1 Client interface

The NodeJS server allows a very fine-grained level of control over the API, and small operations can be called simultaneously when serving pages on the browser. For example, when the project explorer of Dendro is loaded, a series of API calls (via Hyper Text Transfer Protocol (HTTP) request) are made. The results of these calls (all in JSON) are manipulated on the client side (the browser) by an AngularJS app to build the different interface elements. We can therefore say that the web pages are but a shell over Dendro's API, and the Hypertext Markup Language (HTML) rendered by the browser is constructed on the client side. This reduces duplicate code and points of failure, since in Dendro the code behind the API is used for providing information to both humans and machines.

8.5.2 Planning for preservation

Dendro is a completely open-source solution, from its dependencies to the platform itself. It does not require institutions to purchase licensed software to run their own Dendro instance. The technological dependencies were also selected with cloud deployment in mind, since all are designed to easily scale horizontally to respond to the needs of research groups that, albeit being small in size and with access to limited resource, have high demands for data storage. This flexibility to add more or less resources to the platform is important to satisfy the requirements of diverse research groups, as science becomes increasingly driven by the production of large amounts of data (for example, by sensors or other data-generating devices).

From a preservation point of view, Dendro was designed from the start with decommissioning in mind. When the time comes to replace Dendro with another platform, its triple-based data model will help ensure that the data stored within the system lives on. Even after the platform is taken out of service, the semantics of the information stored in the underlying database can be easily understood for migration, since every entity represented within the graph is an instance of a concept from an ontology—this makes the model self-documented, unlike its relational counterparts. Also, in contrast with relational models, whose structure is often influenced by technical factors such as normalization (taking away expressiveness in favor of efficiency and non-redundancy), the graph model exchanges efficiency in some operations for a simpler structure and expressiveness (derived from its ontology foundations). Ontologies were designed from the start to be easily exchangeable on the web through standard representations (such as Web Ontology Language (OWL) and RDF), thus fostering reuse and interoperability between systems—a plus in an environment where the primary goal is to ensure that data survives the deaths of the platforms that hold it.

8.6 CONCLUSIONS

Looking back on the development of Dendro and the choice of a non-relational model for the database, we can say that it was a good choice, because it allowed us to solve the three requirements (Entities with distinct attributes, Hierarchies between resources and attributes whose revision histories need to be recorded) which would be very hard to deal with when using a relational database.

When compared to relational solutions, a graph-based alternative offers several advantages, albeit at the cost of the loss of ACID properties and of the maturity and optimization already offered by the former. From the advantages, we highlight the flexibility for representing relationships between resources with different attributes, less emphasis on the technical aspects of the schema (foreign keys, normalization choices) in favor of domain knowledge when writing queries, as well as the model's ability to be self-documented. The possibility of working in a fully ontology-based environment also eliminates the need to make translations between relational and graph models to provide LOD functionality or SPARQL endpoints. In what concerns Dendro, these features allow us to open the data to the outside world and to let the model grow with the addition of domain ontologies.

When querying a relational database, one must know where to look for the information (which tables to query) and how to get that information (the relationships between the tables). One usually analyses the names of the tables and their columns in order to figure out which of them need to be queried to provide the necessary information. After that is determined, one must figure out the relationships between those tables in order to determine, for example, which tables need to be JOINed to provide all the information required. Given the often abbreviated naming conventions followed in relational databases, the meaning of their tables and columns is usually specified in external documentation, while the relationships are determined through analysis of foreign keys.

Data model documentation is a very important aspect when planning for the future decommissioning of a system. When designing software solutions for use in a long-term preservation workflow, an important goal is to ensure that the data survives the obsolescence of the system that holds them. Unlike relational models, graph-based solutions rely on ontologies for the specification of the semantics of their properties. Ontologies are designed to be publicly shared and reused on the web and their classes and properties can be documented using annotation properties. Thus, we can say that the ontologies themselves document the data model, a very desirable trait in a preservation environment.

In a graph model, an instance of a File or a Folder (concepts from an ontology used to represent directory structures) can also represent an Experiment, a Run, or any data gathering and processing activity carried out by the research group (concepts from domain-specific ontologies). This way, several semantic *layers* can co-exist for faceted querying.

External querying of Dendro's graph can be achieved via the SPARQL endpoint provided by OpenLink Virtuoso—in fact, it is already used by the Dendro platform. Opening the graph to the outside world poses several problems with permission management which were not addressed in the current version. External querying is also not targeted at researchers themselves (as that would require knowledge of the SPARQL language) but rather at software developers intending to build tools designed to support

those external users. As Dendro currently stands, external users can use a free text search to retrieve relevant datasets, but an advanced assisted querying interface (for finding datasets through complex restrictions, for example) is intended as future work. We estimate that it can be implemented rather easily given the simplicity of the platform's data model.

The choice of a technology stack fully supported on JavaScript/JSON in the business logic layer of Dendro enabled us to maintain the code clean and organized. The main issue we encountered was the inexistence of a database abstraction layer for triple stores, and in particular OpenLink Virtuoso. This required the implementation of an "ORM" adaptor from scratch over OpenLink Virtuoso's HTTP API. The result was an additional layer between the database and the business logic, which although making the system slower, allowed the quality of the code to remain high. The database system also supports Open Database Connectivity (ODBC) drivers, so we believe that the performance of Dendro (i.e. number of simultaneous connections, memory usage, etc.) would be greatly increased with the adoption of that alternative technology, as well as with the optimization of the code of our database adapter. Still, we did not encounter any issues related to these concerns when using the current version of Dendro during our tests, so we list these improvements as future work.

9 DESIGN OF LIGHTWEIGHT ONTOLOGIES FOR DENDRO

Ontologies are essential pieces of Dendro's data model, as they allow curators to specify additional descriptors to be used in the description of research data assets. Dendro uses a simpler category of ontologies directly as part of its data model. Existing works call these ontologies "lightweight ontologies" because they are rather simple: they rely on classes and subclasses, properties and subproperties, but leave out axioms and logical restrictions in the definition of those classes. We see them as controlled vocabularies for defining metadata and, as such, they are rather flat when compared to many others commonly found on the web. These other ontologies are much richer than their lightweight counterparts, but their complexity is also the biggest obstacle to their operationalization in a system. We will discuss some of these practical issues in this chapter, as well as the modelling process that we outline for designing new lightweight ontologies for Dendro.

9.1 INTRODUCTION

Research data are often unique and valuable beyond the time frame of individual projects, but their long-term preservation depends on the existence of associated metadata records [36]. Many metadata schemas have been proposed in the past for different domains, often sharing elements with similar or matching meanings [52, 65]. However, their combination or co-existence is not easy, often requiring translation or crosswalk operations [18].

Working around the complexity of standardized metadata schemas, some authors have started to select sets of metadata descriptors suited for their particular application. This "mixing and matching" approach has yielded the notion of *Application Profile* [94]. However, even application profiles can be hard to understand and adopt; moreover, they are self-contained, meaning that they do not evolve incrementally and through reuse like ontologies. For datasets in a fast-paced, multi-domain research environment, a more incremental approach is desirable.

Ontology languages are general-purpose knowledge representation technologies and can be adopted for capturing the nature of metadata records. They convey not only the syntactic rules that enforce the correctness of a metadata record but also the meaning of each descriptor used in a record—in a machine-processable way. They are essential for the description of resources on the Semantic Web [32]. The Linked Data principles rely on a unique identifier, or Uniform Resource Identifier (URI), for each resource on the web, on the existence of links between those resources, and on the ability to retrieve resources linked to a particular one, all of them being

represented using standard formats (e.g. Resource Description Framework (RDF) or Web Ontology Language (OWL)) and queryable via standard languages (e.g. SPARQL Protocol and RDF Query Language (SPARQL)). Generalizing these guidelines to data that is publicly accessible yields the concept of Linked Open Data (LOD) [31].

In this chapter, we present ontologies as an effective way to represent the sets of descriptors that are necessary for the description of datasets from different domains. The ontologies developed according to this methodology are described in more detail in related publications [50, 48, 191]. Each ontology was modeled in close collaboration with a research group from our panel of researchers. They were later loaded into Dendro, our prototype research data management platform, where they were used by the researchers from the different domains to describe several datasets of their choosing.

This chapter is based on the following publications [48, 191, 50]:

- Castro, J., Rocha, J., & Ribeiro, C. (2014). Creating lightweight ontologies for dataset description: Practical applications in a cross-domain research data management workflow. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL and TPDL) DL 2014. ACM Press.
- Rocha, J., Castro, J., Ribeiro, C., Honrado, J., Lomba, A., & Gonçalves, J. (2014). Beyond INSPIRE: An ontology for biodiversity metadata records. In Springer (Ed.), Proceedings of the 10th International Workshop on Ontology Content (OntoContent 2014)
- Castro, J., Perrotta, D., Amorim, R., Rocha, J., & Ribeiro, C. (2015). Ontologies for research data description: a design process applied to vehicle simulation. In Proceedings of the 9th Metadata and Semantics Research Conference (MTSR 2015).

9.2 ONTOLOGIES FOR RESEARCH DATA MANAGEMENT

Ontologies have been presented as a possible solution for research data asset description. They satisfy all desirable metadata requirements [69] and are capable of representing the specific semantics of each research domain, while being able to evolve asynchronously. Yet, the flexibility allowed by ontology modelling processes and the attempts to model every aspect of each domain make it hard to use them in an actual cross-domain research data management environment. OBOE, an ontology for describing and representing ecological observation data, is an example of a domain model whose concepts are very specialized [142]. Like many domain-specific ontologies, its modelling granularity is too fine for it to operationally support a data management system. EXPO, an ontology for scientific experiments [210], is another case of correctness from a modelling perspective, but with a granularity making it unwieldy for usage in an operational data management workflow. Others like ESG [175] and CERIF [122] model cross-domain research concepts representing activities, entities or artifacts relevant for the research workflow that can be used for dataset description—for example *Experiment*, *Project* or *Participant*.

Ontologies with a large number of formal axioms and constraints are sometimes called "heavyweight ontologies", while those with a simpler ap-

proach are called "lightweight ontologies" [129, 55]. Dublin Core (DC), for example, has an OWL version[1] which is currently a widely used lightweight ontology, on par with Friend Of A Friend Ontology (FOAF). Their simplicity and weak constraints make them easily processable by machines, and we have incorporated both directly in Dendro as sources of descriptors. By defining a limited number of classes, avoiding the definition of many object properties, and leaving out constraints and axioms, these ontologies become viable alternatives to support the data model of Dendro—this is because curators can model these ontologies autonomously, helping to grow the data model without having to make changes in the platform's codebase.

9.3 MODELLING LIGHTWEIGHT ONTOLOGIES

To describe their datasets, researchers need to know what metadata to include in the descriptions, something which is usually prescribed by metadata schemas. However, these are often too complex in order to fulfill different metadata requirements [178], making the description process too time-consuming and diverting researchers from their main activities.

We want researchers to add description to data as they are created. The data curators have to provide sets of meaningful descriptors, and to do this we propose the formalization of these descriptors through lightweight ontologies. Building up on past experience we recommend starting the process with a meeting between the curator (a data management expert) and the data creators (domain experts) [49]. During our meetings, we introduce basic data management notions—such as the concepts of metadata and descriptor. After that, we present a set of descriptors from existing metadata schemas. These schemas are generic (such as DC) and research-oriented (such as Ecological Metadata Language [142] (EML) or Data Documentation Initiative (DDI)). After validating a first set of descriptors, the researchers are asked to think about which information is necessary to provide enough scientific context to allow others to verify, replicate, and reproduce the experiments from which the datasets were gathered [179]. After we selected domain-specific concepts to describe datasets, we searched for them in existing ontologies and in controlled vocabularies (for example the Institute of Electrical and Electronics Engineers (IEEE) thesaurus). All the remaining vocabularies that could not be reused from existing vocabularies were introduced in a domain-specific namespace.

9.3.1 An extensible lightweight ontology

Many research datasets consist of sets of files and folders that researchers share within their groups. We have therefore designed a data management platform that uses ontologies to represent such directory structures and relate them to research-specific concepts such as Experiment or Project. The concepts covered in our Research ontology range from the level of the *research experiment* to the level of the *data file*. This means that we do not attempt to represent the semantics of file contents (as is the case of VoID [7], for example), nor the organizational and administrative concepts at the *research project* level (these can be found, for instance, in CERIF). The three ontologies, however, complement each other: Common European Research

1 Available at http://bloody-byte.net/rdf/dc_owl/

Information Format (CERIF) models the highest-level organizational concepts (project-level), Research is targeted at the individual experiments, and Vocabulary for Interlinked Datasets (VoID) is a good approach for modelling the data produced in those experiments.

The Research ontology is therefore focused on representing metadata for *parts of a research project*. This means that, for instance, a *temperature measurement* stored in a file will not be represented with the ontology. But the ontology may include a temperature property to represent the temperature at which an experiment was conducted. An instance of this different (metadata) temperature property can be associated to the *File* or *Folder* of the experiment.

Part **1** of Figure 21 shows the complexity incurred in representing a *temperature* metadata value using two heavyweight ontologies (EXPO and OBOE). Such complexity is undesirable in an operational system in spite of its semantic richness, so we propose a simplified representation via a lightweight ontology (part **2**). We argue that, however, both approaches can co-exist: the metadata of a dataset can be represented using lightweight ontologies in one system and then evolve, via ontology relations, to more heavyweight representations if the need should arise.

The Research ontology models concepts to match the *File-Folder* representation of datasets. It serves as an "extension point" from which other domain-specific ontologies can be derived in order to represent the descriptors required for each domain. It contains concepts such as *Experiment* or *Paper*, that can be reasonably mapped as *Files* or *Folders*. The assumption here is that the directory structure closely follows the different activities of a research project—for example, we consider the Paper concept to represent all the assets and activities in the process of creating a paper, and not the paper as *concrete entity* (unlike EXPO for example). When creating a lightweight ontology for experiments in a specific domain, the curator can also subclass Experiment to create specific types of experiments with their own properties, depending on the domain (see Figure 22).

9.3.2 Instantiating the process

In order to demonstrate the applicability of this process, we have considered two dataset case studies—one from fracture mechanics, and another concerning chemical analysis of pollutant concentrations.

- Double Cantilever Beam (DCB) experiments, in fracture mechanics, consist in testing a given material to study its resistance. A specimen is subjected to pressure so that researchers can evaluate the propagation of cracks in it, recording force values applied and the corresponding specimen displacement. These values are then processed with appropriate methods and converted into energy values.

- Pollutant analysis (PA) is a type of experiment carried out by the analytical chemistry research group. This research group performs routine analysies regarding the concentration of certain pollutants in water and sediments collected *at a given time and place*, in what they call *runs*. These samples are taken and analysed using specific equipment and methods.

Figure 22 shows the lightweight ontologies proposed for the two case studies. The generic Research ontology is shown in **1**, the Double Cantilever

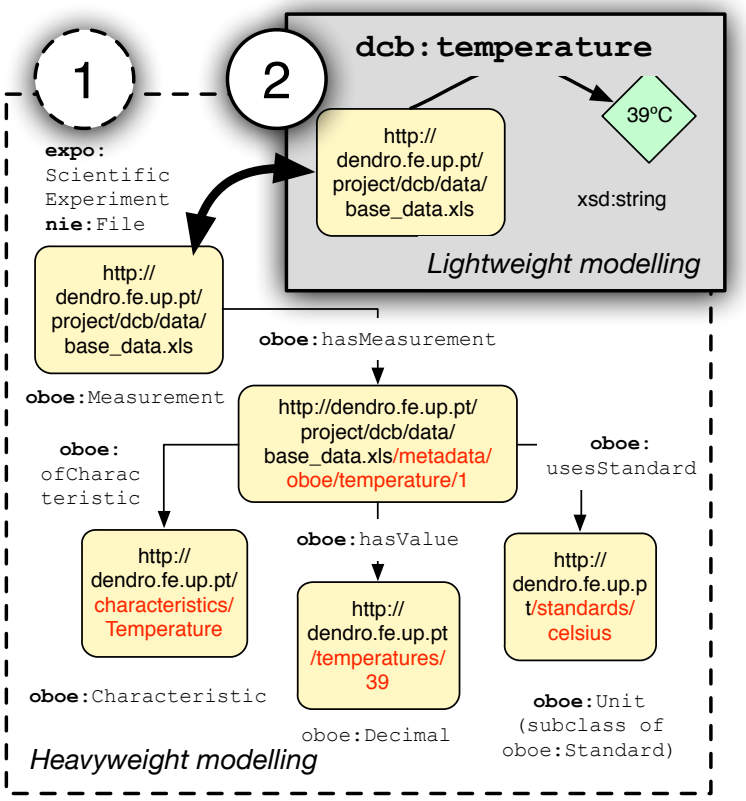


Figure 21: Recording dataset metadata using heavyweight and lightweight ontologies

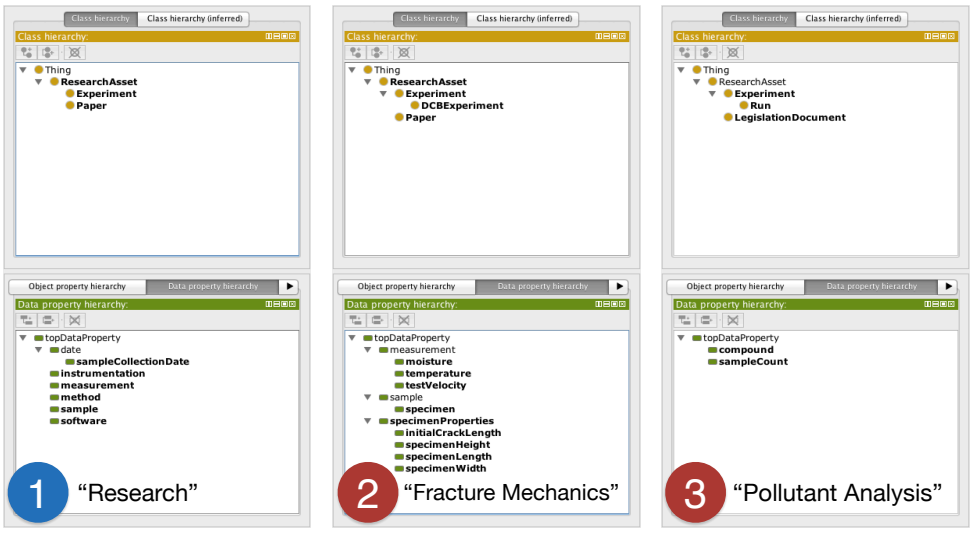


Figure 22: Our lightweight ontologies in Protégé—note the specialization of concepts from **1** in **2** and **3**

Beam (DCB)-specific ontology is shown in **2** and the PA ontology is shown in **3**. The DCBExperiment is derived from Experiment to provide faceted representation (i.e. distinguishing the DCB datasets from the remainder). DCB experiments metadata must include the ambient temperature and moisture

percentage at the location of the experiment, and the velocity at which the specimen was pressed (testVelocity). It is also important to record the specimen that was tested and its properties (specimenLenght, specimenWidth, specimenHeight and its InitialCrackLenght). They are subproperties of specimenProperty that can also be instantiated, allowing researchers to record other metadata.

Researchers from the pollutant analysis domain need to know the number of samples used (sampleCount) as well as the temporal and spatial information of the collected samples. To do so we used concepts from the Dublin Core ontology—namely spatialCoverage—to identify the place where the samples were taken, and specified sampleCollectionDate as a subproperty of DC date. Since this property is cross-domain, it was included in the Research ontology (**1**). Experiments are named *runs* by their creators, so Run was added as a subclass of Experiment and, as researchers compare compound values with legal limits, LegislationDocument (a ResearchAsset subclass) was created to represent relevant legislation (**3**).

9.4 BIOME: AN ONTOLOGY FOR THE BIODIVERSITY STUDIES DOMAIN

Worldwide, environmental changes are acknowledged by their contribution to the increasing rates of biodiversity loss [44, 165, 173]. Understanding and assessing environmental and ecological change is thus essential if the baseline of natural capital is to be defined, in the context of monitoring schemes and long-term ecological research [165, 173]. However, even though the production of spatially-explicit biodiversity data has been increasing, such assessment has been recurrently constrained by incomplete spatial and taxonomical indicator coverage [165].

Biodiversity data often refer to several taxonomic groups, community types, typologies of ecosystem processes, or habitat descriptions across distinct time periods, and are usually collected by distinct groups of researchers. Furthermore, assets of spatially-explicit data related to biodiversity have recently increased significantly, even if they present important differences regarding e.g. their source and methodological development, time of acquisition, and spatial resolution [137, 173]. As a result, there is a pressing need towards the development of common and collaborative networks and tools towards the harmonisation and sharing of data from different sources and scales under common standards and languages [137, 173].

Attempts to tackle such challenges have already been made in the context of specific research projects e.g. the European projects EBONE[2] and BIO_SOS[3], in the context of broader initiatives e.g. the Group on Earth Observation Biodiversity Observation Network (GEO BON)[4]. Specifically, in the case of BIO_SOS project, a geo-portal has been developed so that all researchers from distinct partners and countries could share metadata concerning the datasets produced in the context of the project. The cornerstone of the implementation of this geo-portal was the definition of a metadata schema, built on the INSPIRE core profile, that each of the researchers would fill in during the data gathering stages. This then allowed the evaluation of the fitness of each dataset for the specific intended uses, using a methodology based on

2 http://www.wageningenur.nl/
3 http://www.biosos.eu/
4 http://www.earthobservations.org

the metadata entries, that was itself embedded in the geo-portal [173]. This approach has been used by the InBIO[5] team in recent projects, as a standard toolkit for data management and quality assessment.

The INSPIRE directive 2007/2/EC[6] aims to create spatial data infrastructure for the European Union (EU), enabling the sharing of environmental information among public sector organisations to facilitate public access to spatial information across Europe [25].

The metadata elements specified in the INSPIRE recommendation include the identification of the resource, classification, geographic and temporal references, statements related to the quality and validity of the datasets, conformity with implementing rules on the interoperability of spatial data sets and services, constraints related to access and use, and the organisation responsible for the resource. Information about metadata records themselves is also necessary to ensure that records are kept up to date, and for identifying the organisation responsible for their creation and maintenance.

The INSPIRE directive also considers the possibility of users and systems providing a more detailed description of their resources. This allows them to use additional elements if these are prescribed by other international standards or working practices in their community.

9.4.1 Modeling the BIOME ontology

In the context of Dendro, the collaboration with the InBio reserachers led to the development of BIOME, an ontology that reuses concepts from the INSPIRE directive 2007/2/EC[7], grouped according to the sections of the INSPIRE Geoportal Metadata Editor[8] and complemented by others specified in the BIO_SOS metadata quality guidelines [103].

INSPIRE specifies more than just the metadata representation and exchange formats, it also details how software infrastructures and web services must be designed, for example. Thus, building an ontology for the representation of the entire INSPIRE directive is a very complex task. This model in no way conflicts with other initiatives aimed at capturing the INSPIRE directive in a more fine-grained detail. This is one of the main advantages of modelling with ontologies, the ease with which different approaches can be combined and the incremental development of models at different granularity levels.

All the concepts created for this ontology were given annotations specifying their rdf:labels and rdf:comments. This is important because a natural language description of the concepts is essential to ensure their interpretation by humans [40], and is a good ontology modelling practice overall. Moreover, it is actually necessary to specify these annotation properties to use the ontologies with Dendro, as the platform uses their values to build its dynamic resource description interfaces.

The metadata elements recommended by INSPIRE comprise many categories, including information about the metadata records themselves. This way, metadata are kept up to date and the organisation responsible for their creation and maintenance can be easily identified. The directive also considers the possibility of users and systems providing a more detailed description of their resources. This allows them to use additional elements if these

5 http://inbio.pt/
6 http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32008R1205
7 http://inspire.ec.europa.eu/index.cfm/pageid/48
8 http://inspire-geoportal.ec.europa.eu/editor/

are required by other international standards or the working practices in their communities.

The BIOME ontology is designed for ease of use. It is simple enough to be both easily processable by machines and easily managed by data curators. We have minimized the number of object properties, focusing the model on some core classes and data properties and dispensing with constraints and axioms. The classes capture high-level concepts that represent both data and metadata, while the data properties correspond to the descriptors used in the description of the resources. Resources are therefore captured as instances of the defined classes.

Ontologies that follow this approach have been termed "lightweight ontologies". The Dublin Core (DC) ontology[9] and the FOAF are examples of two widely used lightweight ontologies [55]. We also promote the reuse of concepts from other ontologies, namely DC, FOAF and CERIF[10]. All the relations that we established between BIOME and the DC ontology were based on the "INSPIRE implementing rules for metadata and Dublin Core" [74].

9.4.2 Classes

The first step in the development of the BIOME ontology was to select the concepts to be represented as *classes*. The following concepts were considered essential to capture the top entities in the domain.

1. **Keyword**
   The *Keyword* class captures the concept of a keyword, a special term which can be associated to a resource; the keywords from a specific controlled vocabulary are examples of instances of the *Keyword* class.

2. **GeographicLocation**
   In BIOME this class is defined as a sub-class of the Dublin Core element *Location*, which is in turn a subclass of the *LocationPeriodAndJurisdiction* in Dublin Core as well.

3. **TemporalReference**
   This concept captures the temporal dimension of the data. The INSPIRE directive defines that at least a temporal reference must be provided.

4. **ResponsibleParty**
   The responsible party is an organisation responsible for the establishment, maintenance and distribution of the spatial data and services.

9.4.3 Properties

Table 5 shows the concepts proposed for the BIOME ontology. It illustrates the categories that aggregate the descriptors under the heading **Group**, while presenting the **Properties** in each category—second column. The third column has the **Label** (rdf:label) for each property. The label facilitates the interpretation of the property name by users at the time of metadata creation

9 an OWL version of the DC metadata schema can be found at http://bloody-byte.net/rdf/dc_owl/

10 The Common European Research Information Format (CERIF) Ontology Specification provides basic concepts and properties for describing research information as semantic data. An OWL representation of CERIF is available at http://www.eurocris.org/ontologies/cerif/1.3/

Table 5: Descriptors drawn or adapted from INSPIRE

Category	Property	Label
Identification	resourceTitle(*)	Resource title
	resourceAbstract(*)	Resource abstract
	linkage	Linkage
	identifierNamespace	Identifier namespace
	identifierCode(*)	Identifier code
	resourceLanguage	Resource language
	resourceLocator (≡ dcterms:identifier)	Resource locator
Classification	topicCategory* (⊑ dcterms:subject)	Topic Category
Keyword	keywordINSPIRE(*)	Keyword INSPIRE
	keywordValue(*) (⊑ dcterms:subject)	Keyword Value
	originatingControlledVocabulary:	Originating controlled vocabulary:
	-title*	-title
	-referenceDate*	-reference date
	-dateType*	-data type
Geographic Location	geographicBoundingBox(*)	Geographic bounding box
Temporal reference	temporalExtentStartingDate(*)	Temporal extent starting date
	temporalExtentEndingDate(*)	Temporal extent ending date
	dateOfCreation* (≡ dcterms:created)	Date of creation
	dateOfLastRevision (≡ dcterms:modified)	Date of last revision
	dateOfPublication (≡ dcterms:published)	Date of publication
Quality & Validation	lineage	Lineage
Spatial Attributes	spatialRepresentationType	Spatial representation type
	spatialResolution	Spatial resolution
	spatialResolutionEquivalentScale	Spatial resolution equivalent scale (part of a spatial resolution record)
	spatialresolutionDistance	Spatial resolution distance (part of a spatial resolution record)
Conformity	conformityDegree(*)	Degree of conformity
	specification	Specification
	conformityDate	Conformity date
	conformityDateType	Conformity date type
Constraints	limitationsOnPublicAccess (⊑ dcterms:accessRights)	Limitations on public access
	conditionsApplyingToAccessAndUse (⊑ dcterms:rights)	Conditions applied to access and use
Responsible party	organizationName(*)	Organisation Name
	responsiblePartyEmail(*)	Responsible party e-mail
	responsibePartyRole(*)	Responsible party role
Metadata on Metadata	organizationName(*)	Organisation name
	metadataPointOfContactEmail(*)	Metadata point of contact email
	metadataLanguage(*)	Metadata language
	metadataDate (⊑ dcterms:date)	Metadata date

Table 6: Descriptors drawn from ISO 19115 and GeoCoMaS

Group	Property	Label
GeoCoMaS	projectName	Project name
	version	Version
	diagnosticAndUsability	Diagnostic and usability
ISO19115	distributionFormatName	Distribution format name
	geographicExtentCode	Geographic extent code
	spatialRepresentation	Spatial representation
	referenceSystem:	Reference system:
	-authority	-authority
	-identifier*	-identifier
	-identifierCode*	-identifier code
	-identifierCodeSpace	-identifier code space

in the data management platform interface. Table 6 shows the additional concepts that were drawn from GeoCoMaS[11] and ISO 19115 respectively. In the table, mandatory properties are marked with a $*$ symbol. A $-$ (dash) symbol before the property names in a sequence of properties indicates that the property immediately above them is used as a prefix; this is the case for *title*, *referenceDate* and *dateType* with respect to *originatingControlledVocabularyTitle*. In case there is a subsumption ($\sqsubseteq$) or equivalence ($\equiv$) relationship between a property in BIOME and another from a different ontology, the relationship is expressed in parentheses. For example, *resourceLocator* is equivalent to the identifier property in the DC ontology, so they are marked with ($\equiv$ dcterms:identifier) in their table line.

For the purpose of providing information about the metadata record itself, the properties *metadataLanguage*, *metadataDate* and *metadataPointOfContactEmail* were created. Resource identification is done using the properties *resourceTitle*, *resourceAbstract*, and *resourceLanguage* defined in the ontology. A *linkage* property was also created as a way to record the *resourceLocator* in the form of a URL, along with the *identifier*. The classification of spatial data and services can be described by a *topicCategory* property, that can assist users in finding a resource based on a topic search. The *Keyword* class can be described by a *keywordValue*, that assumes free-text values to represent "a commonly used word, formalised word or phrase to describe the subject"[12], contrasting with the *keywordINSPIRE*, that represents a INSPIRE thematic category, and whose valid values are found in the Gemet Thesaurus[13]. The *originatingControlledVocabulary* specifies valid *Keyword* values and must be "a formally registered thesaurus or a similar authoritative source of keywords"[14], as captured in the *title, referenceDate* and *dateType* of the corresponding controlled vocabulary. For representing the geographic location of the resource, we included the *geographicBoundingBox* property, that must express the westbound and eastbound longitudes and the southbound and northbound latitudes of the intended location. On the other hand the *TemporalReference* class is described with values related to the *temporalEntextStarting/EndingDate*, the *dateOfCreation*, *dateOfPublication* and the *dateOfLastRevision*. We derive most of the temporal and geographical concepts from the DC classes *Location* and *PeriodOfTime*.

11 GeoCoMaS is a normative system of good practices used by the InBIO research team for managing files in their internal repository
12 As described in the INSPIRE online documentation
13 http://www.eionet.europa.eu/gemet/en/themes
14 See footnote 12

To address process history record-keeping and the overall quality of the dataset, along with the *spatialResolution* referring to the level of detail of a dataset, the *lineage*, *resolutionDistance*, and the *equivalentScale* properties were included. To register conformity of the dataset with implementing rules, or other specifications, we define the *conformityDegree* property to represent the degree of metadata conformity, the *specification* it conforms to, the *conformityDate* and the *conformityDateType*.

The constraints related to the access and use are stated via the *conditionsApplyingToAccessAndUse* and by the *limitationsOnPublicAccessAndUse*. Finally, the entity that holds responsibility over the resource must be described using the *organizationName*, the *responsiblePartyEmail* and a *responsiblePartyRole*.

To account for the possibility of a more detailed annotation, which is in compliance with the INSPIRE directive, the BIOME ontology also includes extra elements prescribed by the ISO 19115 [112]. These elements are intended to describe the spatial reference system used in the dataset (*referenceSystemCode*, *referenceSystemAuthority*, *referenceSystemIdentifier*), the digital mechanism used to represent spatial information (*spatialRepresentation*), the description of the geographic area through identifiers (*geographicExtentCode*) and the format in which the data is to be distributed (*distributionFormatName*).

The metadata records created by the researchers in this domain are complemented by elements representing fields of the GeoCoMaS system, and must include a *version* of the resource, the identification of the project where the resource originated from (*projectName*), its domain expressed as a CERIF *Project*, which is in turn a subclass of the *Project* class defined in the FOAF ontology. A summary of data characteristics, quality and usability is specified in *diagnosticAndUsability*.

9.4.4 Using BIOME for describing research data assets in Dendro

The data model of Dendro can be expanded by directly loading additional ontologies into its underlying graph database (using a SPARQL LOAD operation in Dendro's graph database interface), thus providing the flexibility that is expected of a cross-domain data platform. In fact, the platform requires the loading of some ontologies that provide the metadata descriptors before it can run correctly. After loading these ontologies, its properties become available as descriptors for files and folders.

Figure 23 shows more clearly the relationship between the ontology modeling and the application of those ontologies in the Dendro platform, as so far we have only analysed Dendro's interface with descriptors already present (Chapter 8).

The BIOME ontology is based on the analysis of the INSPIRE directive, a relevant standard and the analysis of the metadata records that InBIO researchers already produce (1). After loading BIOME into Dendro (2), its user interface (3) allows the creation of metadata records that include the same descriptors used in the original metadata sheets (see 1A and 3A), but with an ontology-based representation.

9.4.5 Dendro as a description platform for biodiversity datasets

Figure 24 shows a possible integration scenario between the current InBIO information systems and Dendro. Data files and metadata continue to be

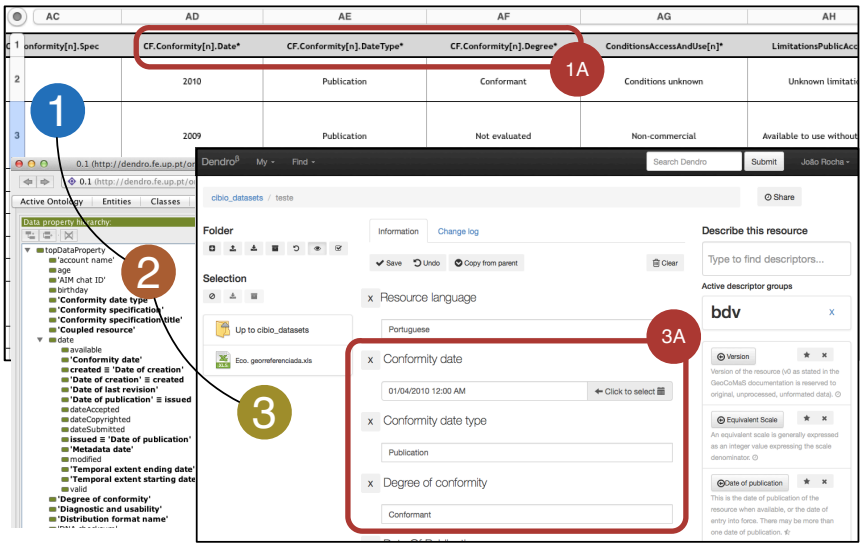


Figure 23: Creating ontology-based metadata records in Dendro

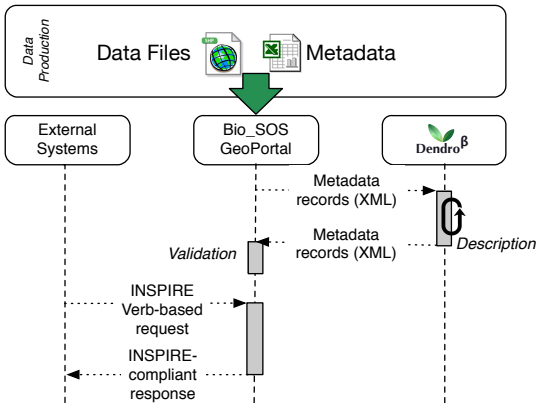


Figure 24: Integrating the current data management platforms of the InBIO team

created and loaded using the geo-portal, with Dendro being regarded as a project-centered data description platform. Dendro can import metadata records in the geo-portal for inclusion in a new project, initiating a process of collaborative description. At any time, Dendro can export the metadata records back on the geo-portal, where they are queried and retrieved using INSPIRE-compatible web services.

9.5 CONCLUSIONS

In this chapter we have demonstrated how to model the *lightweight ontologies* that Dendro relies on for the production of metadata records. Starting from a generic ontology (that we called Research), we have derived several other ontologies for specific domains: analytical chemistry, fracture mechanics and, with a greater detail, biodiversity studies. Highlighting the differences between our lightweight ontologies and those commonly found on the web,

we conclude that, while not as expressive and detailed from a domain modelling perspective, they are easy to implement as part of the data model of an actual application.

Biodiversity information is rich and has many facets to take into account, from very specific ones such as taxonomies of species or community types, ecosystem process typologies or habitat descriptions, spatial and temporal resolution, to more general ones such as geo-referencing, institutional context, time of collection, methods and instruments applied to data collection, and the people involved in it [103].

We integrated biodiversity dataset description in Dendro, which is ontology based, and takes advantage of the set of basic descriptors for biodiversity captured in BIOME, an ontology that includes the relevant properties of data collected by researchers focused on the patterns and dynamics of biodiversity and ecosystems. Building on previous work [173], we have incorporated in BIOME concepts from INSPIRE, an European Commission directive for representing geospatial metadata, to test its applicability in a data management and metadata creation scenario.

Using Dendro and the BIOME ontology, researchers can collaborate in the creation of metadata records that have all the advantages of a fully linked open data representation [190], including SPARQL querying and metadata record representation in formats suitable for LOD de-referencing (e.g. JavaScript Object Notation (JSON), RDF or OWL).

Part IV

Usage-driven application profiles

10 RECOMMENDING DC TERMS DESCRIPTORS VIA A NAIVE RECOMMENDER

Throughout this work, we set out to provide novel ways to help researchers produce metadata records by recommending descriptors based on the interaction with the Dendro platform. One of our first approaches is to identify the types of interactions (e.g. selecting a descriptor from a list, filling it in, hiding it, etc.) that indicate preference for certain descriptors. After determining which interactions should be monitored, a fixed weight for each type is established according to our intuition on their relative importance. The scores of the descriptors are then calculated through a weighted sum, where the number of interactions of different types are multiplied by the aforementioned weights and finally summed up.

In this chapter we describe a first user study where the main goals were 1. to put Dendro through a test with a larger group of users on a real description scenario, and 2. to test this first version of the descriptor recommender. In this study, the recommender was limited to only Dublin Core (DC) descriptors and used a simple ranking formula based on fixed weights.

The study was carried out as an *A-B test* with a group of 23 students of the curricular unit of "Libraries and Digital Archives" of the "Masters in Information Science" at FEUP, and included three weeks of dataset descriptions using Dendro. The results allowed us to conclude that the response to the recommendations was positive overall, with *less effort* per filled-in descriptor, and good feedback being given to various User Interface (UI) parameters of the platform in a questionnaire filled in by the participants.

10.1 INTRODUCTION

In those cases where there is a curation service available to researchers, the data depositor fills in a simplified metadata sheet, that is later translated into a structured representation by a curator [21]; this representation is often made using the Dublin Core Metadata Element Set (DCMES) schema[1] which is widely used due to its simplicity, maturity and interoperability. Since the schema was created before the appearance of Resource Description Framework (RDF), it included no notion of *range* in its specification. The result was that there is some ambiguity in it regarding the possible values for some descriptors. For example, should a string specifying the name of the creator of a resource be a valid value for an instance of a dc:creator descriptor, or should it be the an Uniform Resource Identifier (URI) that identifies the author?[2].

An extended DC specification, DCMI Metadata Terms (DCTERMS), was implemented in 2008 to deal with these issues and introduce finer, more

1 http://dublincore.org/documents/dces/

2 Example drawn from http://wiki.dublincore.org/index.php/FAQ/DC_and_DCTERMS_Namespaces

detailed description semantics. It provided a set of 15 descriptors, specified as sub-properties of the original DCMES descriptors (not to "break" compatibility with existing records), and complemented it with several others. While the improved schema can increase detail in the metadata records, it also adds complexity, since not all elements may be relevant for the description of research products of different nature.

In this chapter, we document a preliminary evaluation experiment carried out using a first version of Dendro that included a descriptor recommender system. The recommendation method was designed around a weighted sum of the number of interaction counts of different types, performed by the users within Dendro. The recommender was restricted to only DCTERMS descriptors, and the test users were 23 students from the Masters course in Information Science at FEUP.

This preliminary experiment served as an effective testing ground for the Dendro platform, with the goal of assessing the robustness of the system as well as its usability. The response of the users to this naive version of the recommender also allowed us to identify some unintended behaviours caused mainly by our fixed weighting scheme, which led to the development of the full recommender, presented in Chapter 12.

User studies involving a relatively small number of users are not the most prevalent in the recommender systems literature, used to dealing with tens of thousands of users and millions of user ratings in their datasets. However, there are some examples where the domain knowledge required from the users makes it difficult to assemble a large user group for performing evaluation experiments. The work of Strickroth et al. [212] demonstrated how even the implicit feedback of even a small number of users can be used to complement a conventional collaborative filtering approach, and how field studies of user satisfaction can complement precision and recall measures in evaluating the performance of the recommender system. Another study where a relatively small user group was used to test a recommender system was presented by Hu et al. [108]; in this study, 30 users participated in a comparison between a personality-based recommender (based on a personality quiz) and a rating-based one. We found the types of questions placed to the users interesting and easy to understand (e.g. "The movies recommended to me matched my interests", to which the user would express different degrees of agreement), so we decided to include similar multiple-choice answers in our survey. Our user feedback survey was also inspired by the work on UI of Sinha et al. [206], with interface issues being divided into several categories—like "Page Layout" and "Use of color" for example. We reused these categories, but presented a 1-5 star scale through which users rated the system categories.

10.2 EXPERIMENT OUTLINE

The requirements expressed by researchers and other stakeholders have been the driving force during the development of *Dendro* and its recommender. Instead of asking the user to spend time rating descriptors from a list, we believed that we would get more genuine user feedback if the systems stood out of the way of the users as they performed their data descriptions. As shown in previous work, such *explicit rating* procedures can disrupt users' *natural* interactions with the system [77], changing the way

they would perform their data description work. We have thus opted for an *implicit rating* model for our recommender.

Given the specific nature of our recommendation task, the low number of test users and the low similarity between their research domains, it was clear at the start of the experiment that the number of ratings gathered would be low. As a consequence, we have decided to outline the evaluation of the proposed approach as a small-scale *user study* instead of an *online experiment*. The absence of a reference dataset ruled out the possibility of carrying out an *offline study* with such a dataset before testing this recommender algorithm with real users in the user study that we now present.

10.2.1 The participants

In order to provide user with a realistic dataset deposit and description scenario, we set up two instances of Dendro at FEUP for this first evaluation test.

We found it hard to gather existing works on user studies where only a small user group was available for the experiment. Small user groups make it hard to establish statistically significant correlations due to the possible bias introduced by the low number of users. This, in turn, limits the conclusions that can be drawn from the studies.

In our usage scenario, users are expected to hold knowledge of data management—including the DC schema—which makes it hard to gather a large user base for our evaluation. In order to draw interesting conclusions in spite of the small number of users involved, we decided to combine a quantitative analysis of a user study with a small user survey intended to evaluate user satisfaction.

Shani et al. [202] state the difficulties that have to be overcome to gather enough users in tasks that can provide interesting and valid conclusions. It is often necessary to split an already small user group into subgroups to compare user performance in different scenarios. We include our experiment in this category, because we decided to split our user group into separate subgroups for *A-B testing*, instead of asking users to perform a part of their tasks in one system and then move to the other.

In our experimental scenario, users could be kept unaware that they were participating in the evaluation of a recommender system. We justify our choice with more realism, which allows us to observe learning patterns in both user sub-groups in order to estimate a learning curve. Some differences between the two user groups were clear, however, introducing a certain bias in the number of descriptors that were filled in at the end of the experiment; one of the groups performed a higher number of annotations overall, indicating that they put more effort into the task than the other.

To test our approach, we had the collaboration of 23 students of the "Digital Archives and Libraries" course at FEUP. All students were already aware of the concepts that are relevant for creating quality metadata, such as the notions of metadata schemas, descriptors and DC. These students are expected to carry out curatorial tasks in their professional environment, which can be a library, a digital repository, an archive, or other system that requires the support of information management experts—in fact, some are already data management professionals.

10.2.2 Cold-start problems

Collaborative filtering is a widely used technique for recommending items to users based on their past preferences. Since we model our users' preferences for the different descriptors by studying their usage, it would seem adequate to apply a collaborative filtering approach—this is not the case, however.

After implementing a prototype using the Apache Mahout library [3], we realized that we would face a serious *cold-start problem*. Without a significant number of interactions by many users over a large set of descriptors, there was a serious risk of rating sparsity, making it impossible to determine similarities between users. Without user neighbourhoods, no recommendations can be produced, and our test users would not use the recommendation features.

Looking at the problem from the point of view of a domain-dependent description task, it is expected that users would describe their datasets using mostly DC descriptors and then complement their descriptions with a few domain-specific ones. This complicates the problem further, because a collaborative filtering algorithm would detect similarities between users due to their common use of DC descriptors, but then would likely present a lot of domain-specific descriptors from domains different from that of the active user.

10.3 DESCRIPTOR SCORING

Instead of providing users with a fixed list of metadata descriptors that users must fill in on every data deposit, Dendro uses information on past user interactions to tailor those metadata sheets to the description needs of the resource and user.

Past interactions over a certain *descriptor* by the *current user* in the context of a Dendro *project* are kept in a log. According to those interactions, each descriptor is given either *bonuses* or *penalties* to its score—Table 7 shows the types of interaction recorded, their description, and the associated *bonuses/penalties*.

The final ranking function used in the system was a weighted sum of these features of each descriptor with some additional changes being introduced for forcefully hiding certain descriptors, for example. In some cases, the number of interactions carried out over the descriptor were also taken into account to multiply the scores up to a certain maximum value.

10.3.1 Testing the recommender for Dublin Core descriptors

Our users' knowledge of information management practices allows us to draw conclusions about how a curator would use the system for depositing and describing datasets. It is often necessary to split the user group into subgroups to compare user performance in different scenarios (*A-B* or split testing), instead of asking users to perform a part of their tasks in one system and then move to the other system to perform other tasks.

In our experiment, the user group was split into two subgroups of 12 and 11 users, that we named U_{Rec} (users of the recommendation-enabled Den-

3 http://mahout.apache.org

Table 7: Weights applied to each descriptor depending on the past interactions of users

Interaction type	*Description*	*Influence in ranking score*
frequently_used_overall	Number of times the descriptor is used over all records in the Dendro instance	Number of times the descriptor was used, up to a maximum of +80.0
from_textually_similar	The descriptor is present in other Dendro resources that are deemed to be textually similar to the one being described (calculated using ElasticSearch's "more like this" feature)[4].	20.0 per occurrence in similar resources, up to a maximum of +80.0.
recently_used	The descriptor has been filled in by the user in the last 30 days	+2.0 per occurrence, up to a maximum of +80.0
used_in_project	The descriptor has been used in the project that contains the resource being described	+2.0 per occurrence, up to a maximum of +80.0
auto_accepted_in_metadata_editor	The user has filled in this descriptor after it was suggested in "Automatic" mode (center area of the interface in Figure 25)	+80.0
auto_rejected_in_ metadata_editor	The user has removed this descriptor from a list as a recommended "Automatic" descriptor in the metadata editor	-80.0
favorite_accepted_in_metadata_editor	The user has filled in this descriptor after it was suggested as a "Favorite" in the metadata editor area, in "Favorites" recommendation mode (see Area **A** of Figure 25)	+80.0
project_favorite	The descriptor has been marked as a project favorite by the user or another collaborator of the current project	+80.0
user_favorite	The descriptor has been marked as a personal favorite by the user	+80.0
project_hidden	The descriptor has been marked for hiding in the current project	N/A (Force hiding, regardless of score)
user_hidden	The descriptor has been marked as a personal hidden descriptor by the user	N/A (Force hiding, regardless of score)

dro) and U_{Alpha} (users of the Dendro that presented descriptors ordered alphabetically). Each of the two subgroups was further divided into teams of 3 students when possible to suit the method used in the course, which follows a realistic scenario usually encountered in curator teams. The 3-person teams were selected by the students themselves and then attributed to U_{Rec} or U_{Alpha} randomly.

Each of the teams was tasked, over three weeks with the description of several datasets from different online sources that we selected. The teams were allowed to pick datasets from different research domains and natures, as long as they were drawn from the online sources that we selected.

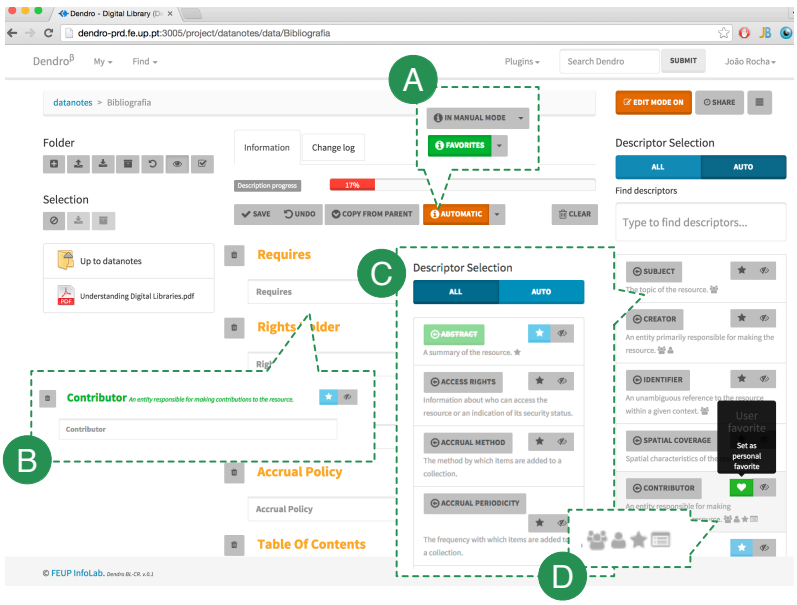


Figure 25: The main user interface of Dendro

Figure 25 shows the user interface of Dendro, with the recommendation features highlighted. The layout is divided in three main sections, from the left to the right: the file browser, which allows users to navigate in their files and folders; the metadata editing area, which presents all the descriptors and controls for users to specify their corresponding values (text boxes, date pickers or maps); finally, on the rightmost position there is the descriptor selection area where users can select descriptors to be included in the current metadata record. Area **A** shows the possible *interface modes* that users can switch between, in order to produce their metadata records. When the "Manual mode" option is selected, no descriptor suggestions are automatically added to the centre area; when the user switches to the "Automatic" mode, a set of descriptors recommended by the platform are automatically added to the description area in the centre of the user interface, if they are not already filled in for the current record. In "Favorites" mode, descriptors that were marked as *project favorites* or *user favorites* are automatically added to be filled in (if they are not already present).

The interface highlights descriptors suggested via the "Automatic" mode in yellow and descriptors added via the "Favorites" mode in green (**B**). The descriptors selection area (**C**) allows users to add descriptors to the metadata record by clicking the button in each row. For each descriptor in this area, two additional buttons are present: one to *promote* a descriptor to *fa-*

vorite, and another to *hide* that descriptor, serving as a descriptor ranking functionality.

Upon clicking the *favorite* button for the first time, the selected descriptor is marked as *project favorite* for all the project collaborators in the context of that project (highlighted in a light blue and with a star badge). The second time the button is pressed, the descriptor will, in addition, be promoted to *user favorite*—meaning that the descriptor becomes a favorite for that user, being recommended as such in the context of any project that the user is a part of, but without appearing on the descriptor recommendations of the other project collaborators.

While the user that marks a descriptor as a favorite will always see the their favorite at the top, project collaborators that did not favorite that descriptor will not necessarily witness the same behaviour; this is because, for these users, the descriptors are only given a strong bonus that can be overcome if enough other favorable interactions towards other descriptors are performed in the system.

Descriptors marked as *hidden* by the users are immediately hidden from the current user's view and also heavily penalized overall, making it harder for them to be recommended to other users in the same project. Favorite descriptors, on the other hand, are benefitted with a strong bonus to ranking whenever an user marked them as a project favorite. This made them appear high on the ranked list for the users that also participated in the same project.

Dendro provides feedback to the user regarding the reasons why a given descriptor was included in the list of recommended ones. Under each of the buttons that users can press to include a recommended descriptor in their metadata record (Area **B**), a set of icons can appear to indicate the factors that played into the recommendation of that descriptor (e.g. "Frequently used in project", "Frequently used in the entire platform", "Used in textually similar resources", etc.). These explanations improve system *transparency*, which has been shown to improve the effectiveness of a recommender system [207, 214]. In the specific case shown in Figure 25, the icons indicate to the user that the descriptor was chosen because it has the following bonuses applied to its ranking: it is frequently used by Dendro users ("group of people"); it was used by the user ("single person"); it is a project favorite ("star"); it was used in the project ("check list").

The interface elements in areas A, B and D are only available in the D_{Rec} instance. The C elements is available on both Dendro instances, but the "Auto" button was not visible, with only the list of descriptors being shown for users to perform their selections.

10.4 CONCLUSIONS

This first experiment enabled us to put Dendro through a user test, proving its ability to support several users working simultaneously on the platform, with little technical problems occurring during the test sessions. The users expressed positive feedback on the overall user interface and the robustness of the solution, and were able to carry out the proposed tasks.

This descriptor recommender for DC descriptors was a preliminary experiment with preliminary results—however, when we consider that most repositories base their metadata model on DCTERMS, we can say that this approach was at the level of those existing solutions.

This test experiment was important for us to discover several necessary improvements in order to make recommender suitable for multi-domain descriptor recommendation and at the same time help us identify potential pitfalls in the design of the final experiment. For example, the lack of normalisation in the scores made them have sometimes incorrect behaviours. The fixed limits placed on the score components were an attempt to deal with this, but ultimately they would be overcome after enough interactions of lower importance were performed.

In the next chapter we explain how we took the lessons learned from this test to produce the final version of the recommender algorithm. Among other necessary improvements, we replaced the fixed weights on the interactions weighting schema with a parametrizable ranking formula, introduced new types of interactions and, of course, domain-specific descriptors in addition to DC ones.

11 IMPROVED DESCRIPTOR RECOMMENDER ALGORITHM

This chapter documents the changes made to the descriptor recommender system after the first experiment. The improved version was designed to deal with domain-specific descriptors from multiple domains, while the ranking algorithm was refined, including score normalization, a new scoring formula and additional interaction types.

11.1 INTRODUCTION

Dendro approaches the problem of metadata record production by assisting the researchers in the process of data description. Empowering these users, which are not experts at metadata production to create their own metadata records, requires the definition of descriptors from concepts of their own research domains. To parametrize Dendro, curators can specify ontologies which define the sets of descriptors that the researchers will have at their disposal for describing datasets, as described in our previous works [190, 189].

The platform is designed to allow institutions and groups to load as many descriptor-defining lightweight ontologies as necessary, and then provide an environment for researchers to select and fill in the descriptors that they consider relevant for their datasets or domains. By selecting these descriptors and filling them in, researchers implicitly collaborate to compose an application profile [94] for the research group.

This approach has a problem, however: as more and more ontologies are included to provide the descriptors required by different domains, it becomes harder for researchers to select the interesting descriptors from among an increasingly large number of available ones. This inability to choose between an overwhelming set of alternatives is designated *information overload* in recommendation [184].

Dendro approaches this problem by helping researchers in the task of descriptor selection. The approach is based on recording many kinds of interactions performed by users in the system. This information is used to train a ranking algorithm, which can then recommend lists of descriptors that researchers may be interested in filling in, in order to describe the contents of a given file or folder, therefore producing metadata records tailored to the needs of each research group.

The available descriptors (*items* in recommender systems terminology) are evaluated according to a set of characteristics (*features*) and then ranked according to their score on the features. An effective ranking can be obtained with a combination of the feature values, where the relative importance of the different features is taken into account. In our case, the items are descriptors, and their features are derived from counts of user interactions grouped by kind, and which result from descriptor usage. We estimate the importance of different interactions over a descriptor according to our intuition on the domain, in order to devise a convenient weighting schema.

For example, the act of filling in a descriptor expresses some preference towards that descriptor, but setting a descriptor as a *Favorite* expresses a much stronger preference.

In this chapter we document how we devised the ranking algorithm used in the second and final user study carried out to prove our hypothesis. We start by clarifying the reasons why we opted for a ranking approach instead of a more common recommender algorithm, such as content-based recommender systems or collaborative filtering. We then outline the behaviour required of the scoring function, the normalization method applied to the features, the scoring function itself, and present a scoring example.

11.2 REASONS FOR ADOPTING A RANKING APPROACH

The main reason for adopting a ranking approach instead of a content-based recommendation algorithm is the variety of research dataset in existence, which can make it impossible to gather any useful information only by looking at their content. Another alternative, the collaborative filtering approach, would be more suited to cases where there is a large number of users and ratings the *cold-start* problem. The cold-start problem refers to the inability of the system to provide recommendations to users that have not rated any items in the system. The expression can also be applied to the items, designating the inability to recommend items that have not been rated by any user in the system [200]. When considering item-item similarity in content-based recommender systems, the term *cold-start* is also designated as the *first-rater problem* [149]. In our case, we expected to have an insufficient number of users overcome the cold-start, because we had to gradually gather our user groups and manually design or gather the necessary ontologies for them to describe their datasets.

We have previously made some exploration of a content-based method [194], which recommended descriptors based on the most important terms present in texts extracted from documents such as Portable Document Format (PDF) or Word files. This approach can be useful at a later stage, in a context where lots of links between datasets, papers, research groups and projects are available; for now, we do not know which types of files are brought in by the researchers. Since we we want to avoid a situation where we would be unable to recommend anything because the files would be purely numeric [196, 193], we opted for analysing user interactions instead of file contents, thus ensuring that we always have data to base our recommendation on.

11.3 INTERACTION RECORDS

As users interact with the system, selecting descriptors and filling them in, several aspects of the interaction are recorded: a *timestamp* for the interaction, the user that performed it, the kind of interaction (e.g. selecting, filling in, hiding, setting as a favorite, etc.), the descriptor over which the interaction was performed and the file or folder that was being accessed when the interaction was performed. Since a file or folder can only belong to a single project, we can also determine all the interactions performed over all the files and folders of a project by analysing these records.

The position on the list (or *rank*) of the descriptor at the time when the interaction is performed is also recorded, if that is relevant for that particular interaction type. These rank records constitute implicit feedback on the effectiveness of the ranking, to be used in the evaluation of the recommendation. For example, a descriptor that is accepted from a top position in the list required less effort on behalf of the user than a descriptor lower down, because the user did not have to scroll to find it.

11.4 IMPLICIT VERSUS EXPLICIT FEEDBACK

Users can express their preferences for the items presented to them through *implicit* or *explicit* feedback [158].

While explicit feedback is gathered through direct rating by the user such as like/dislike, rating on a 1-5 scale, or even continuous scales [214], implicit feedback is gathered after analysing certain user actions and combining those with knowledge of the domain. In web retrieval, some of the most important sources of implicit feedback are the *click-through*, accounting for users selecting a given result [117], *dwell time*, the time users spend viewing a web page after selecting it from a list of results, as well as their scrolling behaviour [158], *eye-tracking* [117], or a combination of many signals [2]. Our solution uses mostly *implicit* but some *explicit* preferences are also recorded. For instance, selecting or filling in a descriptor is an implicit positive feedback of the user about a descriptor, while promoting it to a *Favorite* constitutes explicit feedback on the relevance of the descriptor to the user.

It is well known in recommender systems that working with implicit feedback allows the gathering of more information, and of a more diverse nature, than explicit feedback can provide. This helps to deal with the cold-start problem.

It is also necessary to transform implicit feedback signals obtained from usage information into numerical ratings which can be used in recommendation models. For example, the time interval spent viewing a web page can be turned into a value on a scale from 1 to 5. This mapping requires some consideration on the nature of the signal and the actual distribution of its values. A good understanding of the domain for which the recommender system is being built is required to devise a mapping between implicit signals and values to be used in a ranking algorithm [212]. In our case, our experience in working with researchers throughout the development of Dendro and the experience provided by past studies forms the basis for the proposed model.

11.5 DESCRIPTOR FEATURES

We have defined a set of features to be used in descriptor recommendation based on the available usage information. They are defined for each descriptor and are mostly based on the accumulated sums of interactions of several kinds. Table 8 lists the features (first row), and the types of interactions contributing to each (first column). A ✔ is displayed whenever the corresponding type of interactions is included in the interaction count for a feature. As an example, the "Frequently selected overall" feature is calcu-

lated by accumulating the counts of the interactions of the types listed in the first 5 rows, plus those of the type in the 8th row of the table.

The first column classifies each interaction as an *effort* or *non-effort* interaction. Every descriptor selection that is performed by the user clicking a button, for example, is an effort interaction. Other examples of effort interactions are clicking one of the descriptors in a list, browsing to the next or previous page of descriptors, or selecting an ontology to access the list of descriptors therein.

In contrast with effort interactions, any logged interaction that results in the addition of a descriptor to the metadata editor without the direct action of the user is considered a non-effort interaction. Examples are the automatic inclusion of a descriptor in the metadata editor while Dendro is in *Auto* or *Favorite* descriptor selection mode; these modes are selected by the user (as shown in areas **2a** and **3a** of Figure 28). It is important to mention that this type of non-effort interactions are only logged when the descriptor is actually filled in after it is added to the editor, or else the system would log many meaningless non-effort interactions resulting from web page reloading.

The features are ordered by ascending importance, corresponding to the observation of user behaviour. Some interactions are more important than others because they typically result in more useful actions, or they convey some strong intention by the user. For example, selecting a descriptor from the list expresses some degree of preference for that descriptor; however, if the user actually fills in the descriptor, that interaction indicates a stronger evidence of positive preference of the user towards that descriptor. Setting a descriptor as a *favorite* expresses an even stronger positive preference of the user towards that descriptor—setting a descriptor as favorite constitutes *explicit feedback* from the user, who must stop performing their description tasks to click the Favorite button.

11.6 NORMALISATION OF INTERACTION COUNTS

Interaction counts are accumulated, and due to their nature the sums can have different ranges. We use them as signals that will have different impact in the ranking formula. To consider the signal and be immune to the value range, interaction counts are normalised. Each is divided by the maximum value registered for the corresponding feature. This is known as *min-max normalization*, and has the benefits of retaining the relationships between ratings while being very simple to calculate [90].

11.7 DESCRIPTOR RANKING

A weighted sum can provide a simple score for a descriptor. If v_{id} is the normalised score of feature i of descriptor d, and w_i the weight assigned to feature i, the sum yields the score of descriptor d:

$$\text{Score}(d) = \sum_{i=1}^{n} w_i \cdot v_{id} \tag{8}$$

Table 8: Different types of interactions recorded by the system

Effort?	Interaction type \ Feature	Frequently selected overall	Frequently used overall	Favorites overall	Frequently selected in project	Frequently used in project	Frequentl y used by user	Frequently used by user in project	Project favorite	User favorite	*Hidden*
		Importance →									
Yes	Select a descriptor from the list of recommendations	✓			✓						
Yes	Select a descriptor from the manual list of descriptors	✓			✓						
Yes	Select a descriptor from the manual list, at a point in time when the descriptor was marked as a project favorite (descriptors can be marked and unmarked as favorite, so it is valid at the time of descriptor acceptance)	✓			✓						
Yes	Select a descriptor from the manual list, at a point in time when the descriptor was marked as a favourite by the user.	✓			✓						
Yes	Select a descriptor from the manual list, at a point in time when the descriptor was simultaneously marked as favorite by the user and favorite in the project	✓			✓						
No	Fill in a descriptor automatically added to the metadata editor while the "Automatic" mode of metadata editing was active		✓			✓	✓	✓			
No	Fill in a descriptor automatically added to the metadata editor while the "Favorites" mode of metadata editing was active		✓			✓	✓	✓			
Yes	Mark a descriptor as "User Favorite" while it is presented in the list of recommendations			✓					✓	✓	
Yes	Unmark a descriptor as a "User Favorite" while it is presented in the list of recommendations			✓					✓	✓	
Yes	Mark a descriptor as "Project Favorite" while it is presented in the list of recommendations			✓					✓	✓	
Yes	Unmark a descriptor as "Project Favorite" while it is presented in the list of recommendations			✓					✓	✓	
Yes	Fill in a descriptor which was added to the metadata editor area by a click on one of the descriptors in the manual list, but was neither a User nor Project favorite at the time		✓			✓	✓	✓			
Yes	Fill in a descriptor that was selected from the manual list, at a moment when it was a User Favorite		✓			✓	✓	✓			
Yes	Fill in a descriptor that was selected from the manual list, at a moment when it was a Project Favorite		✓			✓	✓	✓			
Yes	Fill in a descriptor that was selected from the manual list, when it was both Project and User Favorite		✓			✓	✓	✓			
Yes	Fill in a descriptor that was added to the metadata editor from the list of recommendations by clicking on it (neither User nor Project favorite at the time of selection)		✓			✓	✓	✓			
Yes	Fill in a descriptor added from the list of recommendations while it was a User favorite		✓			✓	✓	✓			
Yes	Fill in a descriptor added from the list of recommendations while it was a Project favorite		✓			✓	✓	✓			
Yes	Fill in a descriptor added from the list of recommendations while it was a User and Project favorite		✓			✓	✓	✓			
No	Fill in a descriptor that was inherited from the parent folder in the project using of the "Copy from Parent" button in the metadata editor and then saving. One interaction recorded per "inherited" descriptor.		✓			✓	✓	✓			
Yes	Hide a descriptor from the list of recommendations, will mark it as Hidden for the current project. Will not show up anymore in the recommendations or manual list until the Hidden status is removed by a project member										✓
Yes	Reverse the Hidden status of a descriptor in the current project (will once again show up in the project's recommendations and be available in all files and folders of the current project)										✓
Yes	Hide a descriptor from the list of recommendations, will mark it as Hidden for the current user, regardless of the current project. Will not show up anymore in the recommendations or manual list until the Hidden status is removed by the user										✓
Yes	Reverse the Hidden status of a descriptor for the current user (will once again show up in the users' recommendations and manual list)										✓

It became apparent in preliminary studies that the weighted sum would not be able to capture all the aspects of our empirical observations concerning the impacts of features on descriptor ranking. The features listed in Table 8 are different in nature, and we want to value their relative importance and also generate useful recommendations even in the early stages of Dendro adoption, when the number of users is not large. We have therefore decide to compute scores based on a family of weight function rather than on fixed weights. A function can be equally easy to calculate, and allow a much more flexible behaviour of the feature scores with respect to the feature values.

To define the shape of the scoring functions, we identified a set of properties, and them selected a family of functions which enjoy these properties. One aspect that we want to emphasise is the importance of a feature coming into play, and this can be strengthened with a sharp transition from a zero to a non-zero value. We also want to make the ranking respond quickly to just a few interactions of the users. The ranking function is designed to emphasise the first interactions of each type, for each descriptor: the first few interactions should have the largest impact in the score component for that particular type of interaction. Conversely, as more interactions of a type are recorded for that descriptor, the lower the difference they will make in the score.

Since our normalised values range from 0 to 1, we looked for a family of scoring functions with the following properties:

1. Non-negative in the $[0, 1]$ domain—all our features convey positive information;
2. Non-linear behaviour in 0—mark the transition from 0 to a non-zero feature value;
3. A high slope for values of x close to 0—emphasise early interactions;
4. Slope tends to 0 as the values of x approach 1—dampen the importance of additional interactions as they approach the maximum value;
5. Strictly ascending in the whole $[0, 1]$ range—a higher number of interactions yields a higher score;
6. Flexible range—score ranges $]f(0); f(1)]$ are set according to the importance of the feature;
7. 0 score for a feature value of 0.

Analysing these properties, we have considered a cubic curve as the scoring formula for our features. $f_i(x)$, the score function for feature i, where x is the normalised value for feature i, is then defined as follows:

$$f_i(x) = \begin{cases} 0, & \text{if } x = 0 \\ r_i(x-1)^3 + f_{i_{max}}, & \text{if } 0 < x \leqslant 1 \end{cases} \tag{9}$$

This definition separates the case when the normalised count of relevant interactions for feature i is 0, enforcing the obvious 0 value for the feature score. This allows for the function to have a nonlinear behaviour in 0: for a nonzero feature value, the function steps to $f_{i_{min}}$ to allow for a strong contribution for any nonzero feature value.

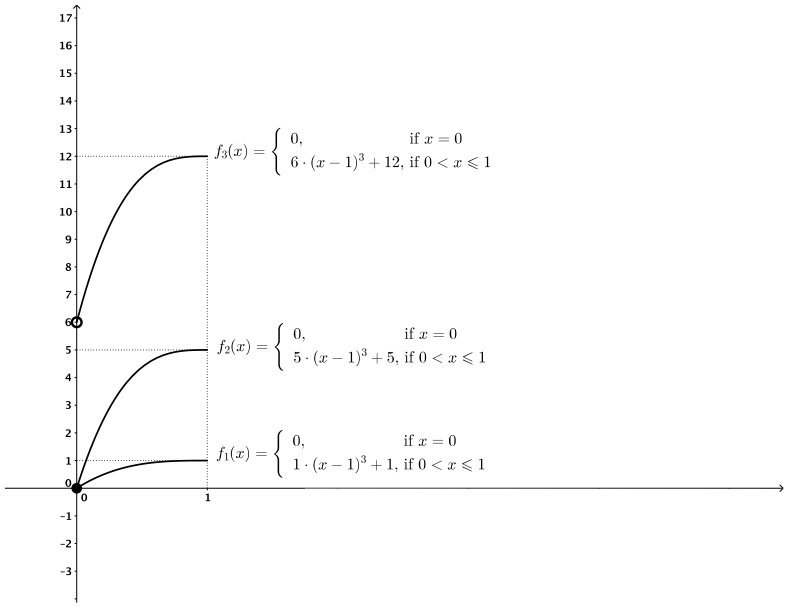


Figure 26: Examples of two descriptor scoring functions

For nonzero values, the function takes values that range from $f_{i_{min}}$ and grow monotonically with the feature value, with decreasing slope, to $f_{i_{max}}$. This means that small values of the feature have a relatively larger impact on the score than large ones. The range of the nonzero feature score values is r_i in Equation 9, and therefore

$$r_i = f_{i_{max}} - f_{i_{min}} \tag{10}$$

We have set the values of $f_{i_{min}}$ and $f_{i_{max}}$ for each feature empirically, according to the relative importance we estimate for each feature. The cubic function satisfies the above requirements, providing an evolution for the score with an initial value of $f_{i_{min}}$, in case the feature is present, and then evolving to $f_{i_{max}}$ with a faster growth on small values. Figure 26 shows the scoring functions for three features—notice how the changes in the $f_{i_{max}}$ and $f_{i_{min}}$ parameters change the contribution of the feature for non-zero values as well as the range of values it may assume.

We can now combine the feature scores into a general score for each descriptor. v_{id} denotes the normalised value of feature i for descriptor d, and f_i is the scoring function for feature i according to Equation 9. $\mathrm{Score}(d)$, the score of descriptor d, sums the contribution of each of the n features of d:

$$\mathrm{Score}(d) = \sum_{i=1}^{n} f_i(v_{id}) \tag{11}$$

11.8 PARAMETER SELECTION

According to our knowledge of the domain and observation of the researcher behaviour, we have set the parameters for each feature scoring function. Table 9 shows the $f_{i_{min}}$ and $f_{i_{max}}$ values for each feature, the corresponding

Table 9: Parameter setup and ranking formula for every feature

Feature	**Description**	**Min Score** ($f_{i_{min}}$)	**Max Score** ($f_{i_{max}}$)	r_i	**Scoring Formula**
f_1	Frequently selected overall	0	1	1	$1 \cdot (x_i - 1)^3 + 1$
f_2	Frequently used overall	0	5	5	$5 \cdot (x_i - 1)^3 + 5$
f_3	Favorite overall	6	12	6	$6 \cdot (x_i - 1)^3 + 12$
f_4	Frequently selected in project	17	34	17	$17 \cdot (x_i - 1)^3 + 34$
f_5	Frequently used by user	46	92	46	$46 \cdot (x_i - 1)^3 + 92$
f_6	Frequently used in the current project	126	252	126	$126 \cdot (x_i - 1)^3 + 252$
f_7	Frequently used by the current user in the current project	344	688	344	$344 \cdot (x_i - 1)^3 + 688$
f_8	Current user favorite	940	1880	940	$940 \cdot (x_i - 1)^3 + 1880$
f_9	Current project favorite	2568	5136	2568	$2568 \cdot (x_i - 1)^3 + 5136$

range r_i and the complete expression for each function. The least important features (1 and 2) start at the minimum of 0, but the maximum value of the least important—*Frequently selected overall*—is 1, while the one immediately above can go up to a maximum value of 5, overtaking the least important one if enough interactions exist, but not overriding it from the start. This case is illustrated in Figure 26 by the two curves for f_1 and f_2.

Overall designates the features that count interactions over all the files and folders in the platform and regardless of the user that performed them. From the three *overall* features, f_3 or *Favorite overall*, starts at a value of 6. It is the first feature derived from explicit feedback and we configured it to override the two previous features which are based on implicit feedback and thus require less user effort. The maximum value is then specified as twice the minimum value, for f_3 and above.

$$f_{i_{max}} = 2 \cdot f_{i_{min}} \tag{12}$$

For f_4 and above, the minimum value of the score for feature i is the sum of the maximum values of the two features immediately preceding it. This ensures a steady increase in the weights as the features grow in importance.

$$f_{i_{min}} = f_{(i-2)_{max}} + f_{(i-1)_{max}} \tag{13}$$

Figure 26 illustrates the weighting behavior in the f_3 curve, which always has a larger value than the sum of f_1 and f_2.

11.9 RANKING EXAMPLE

Table 10 shows a score calculation example following the proposed algorithm, using only 4 descriptors and 5 features. In this table we have four descriptors to be scored: two from DCMI Metadata Terms (DCTERMS) (dcterms:title and dcterms:subject), one from Friend Of A Friend Ontology (FOAF) (foaf:mbox) and finally one from the Double Cantilever Beam (DCB) ontology (dbc:initialCrackLength). DCB stands for "Double Cantilever Beam"—it is a domain-specific ontology designed to handle the specific description needs of a research group working in the domain of Fracture Mechanics at Faculdade de Engenharia da Universidade do Porto (FEUP). The dcb:initialCrackLength descriptor documents the initial crack made to a specimen during an experiment designed to study its displacement when subjected to an increasingly large force, until it finally fractures.

Features become more important as we move from right to left in the table. Thus, we first need to compare the rightmost features; if a descriptor has a non-null value for that feature and the others do not, that descriptor should rank above the others. In case of a score tie, we then move left and compare the scores of subsequent features.

Descriptor dcb:InitialCrackLength is marked as project favorite, and so is dcterms:subject. This is the feature with the highest importance. The other two (dcterms:title and foaf:mbox) have a zero value for that feature, so we already know that they should be scored lower.

To decide which one of the two project favorites will be ranked higher, we look left along the feature columns; the next feature is *User Favorite*, and both are zero on this one, so we cannot take a decision yet.

Table 10: Parameter setup and ranking formula for every descriptor feature

Descriptor \ Feature	Frequently selected overall	Frequently used overall	Used by the current user in the current project	Current user favorite	Current project favorite
Step 1 - Interaction counts					
dcterms:title	312	254	3	1	0
dcterms:subject	41	38	1	0	1
foaf:mbox	12	11	1	0	0
dcb:initialCrackLength	5	5	4	0	1
Max(Feature)	312	254	4	1	1
Step 2 - Normalising by max					
dcterms:title	1	1	0,75	1	0
dcterms:subject	0,13	0,15	0,25	0	1
foaf:mbox	0,04	0,04	0,25	0	0
dcb:initialCrackLength	0,02	0,02	1	0	1
Step 3 - Scores according to ranking formula (Equation 9)					
dcterms:title	1	5	682,63	1880	0
dcterms:subject	0,34	1,93	542,88	0	5136
foaf:mbox	0,11	0,62	542,88	0	0
dcb:initialCrackLength	0,05	0,29	688	0	5136

Descriptor	Score	Final rank
dcterms:title	2568,63	3
dcterms:subject	5681,15	2
foaf:mbox	543,61	4
dcb:initialCrackLength	5824,34	1

The next feature is *Used by current user in the current project*, and for this feature we have a descriptor with non-zero value (dcb:initialCrackLength, value 4), while the other (dcterms:subject) has a value of 1—this means that dcb:initialCrackLength should rank higher in this pair.

We have that dcb:initialCrackLength should rank first, and dcterms:subject rank second. For the other two descriptors, we follow the same reasoning: since none of them are *Project favorites*, we check the values of the features immediately below in importance in the scale, *User favorite*. Since dcterms:title is a user favorite (value 1) but foaf:mbox is not (value 0), dcterms:title should be ranked higher. Since we already know which descriptors rank first and second, the final rank will be:

1. dcb:InitialCrackLength
2. dcterms:subject
3. dcterms:title
4. foaf:mbox

We highlight how an interaction count of 1 or 2, when part of a more important feature, can cause a high impact on the final score, while hundreds of counted interactions from a lower-weighted feature can result in a lower contribution to the score. This is intentional and designed to make certain features override others, as seen in Figure 26, where the weights are calculated so that a more important feature (f_3) can be reinforced to have a value larger than the sum of the those with lower importance (f_1 and f_2).

The steps of the calculations for this example are shown is Table 10. In Step 1, we gather the counts of those interactions relevant for each feature, as specified in Table 8. After applying the scoring functions over the features, we sum them all for each descriptor and rank them in descending order. The final score results as expected.

11.10 CONCLUSIONS

Our proposal for a descriptor ranking algorithm uses interaction counting as the main source of evidence of user preference towards certain descriptors. In this recommendation algorithm, interactions have different types, which are given varying weights according to their importance. The ranking algorithm itself is a sum of components. These components, the descriptor *features* are calculated via a non-linear function which has the normalised values of different interaction counts as its independent variable.

When compared to the ranking formula outlined in our initial experiment, this algorithm deals with the numeric bias introduced by the lack of normalization, while allowing easy parametrization of the ranges ranges that the different features can assume.

In the absence of a baseline study of this type, we had to devise a reasonably simple solution capable of providing adequate recommendations, even in the absence of large numbers of user ratings as it is commonly found in recommender systems literature. This is because of the very specific nature of our target user groups, which were all active high-level researchers. The specific nature of the task itself (data description) also led us to believe that it would be hard to gather a significant number of users to participate in the

testing stages. We consider this ranking formula to be a baseline over which others can improve as well.

The ranking formula that we chose is by no means the only possible alternative. In other ranking works, for example in Information Retrieval (IR), logarithmic functions are used to express dampening as scores increase [205]—this is another possibility for experimentation in the future.

In the presence of larger user groups, more commonly seen alternatives used in recommender systems could have been used—such as collaborative filtering, for example. More users mean more files and datasets, and content-based approaches could also have been introduced through automatic text extraction from documents as they are uploaded to Dendro. We have introduced text extraction as a part of Dendro's free text search capabilities, so the foundations for an approach based on important term extraction, as presented in our past work [194] are already in place.

The next chapter documents the user study that we carried out in order to test our hypothesis in a real data description scenario with researchers from various research areas.

12 USER STUDY WITH DOMAIN-SPECIFIC DESCRIPTORS

This chapter documents an evaluation experiment carried out with 11 research groups from the University of Porto where Dendro was used as a data management environment which included our descriptor recommender. Our panel includes elements from very diverse domains, from Engineering to Social and Behavioural Sciences, who participated not only in the data description experiments themselves, but also in the design or selection of the ontologies that were used.

The preliminary results or our evaluation show that usage-based descriptor selection can be a good complement to the conventional approach based on separate metadata schemas; the results also show that users fill in more domain-specific descriptors when the recommender system is in place; finally, metadata records of similar quality (as judged by a data curator) are created with less effort when the recommender is enabled.

12.1 INTRODUCTION

Our contact with researchers in several areas for a period of over 4 years showed us that researchers can be motivated to data description, but do not have enough motivators (or penalties) to dedicate a significant part of their high-valued research time to data description. Recommendation is aimed at a less committed, more *side-effect* attitude on their part. If we can do a good enough job with metadata resulting from a rather uncoordinated, time-distributed effort, this will be an excellent indication in favor of data publication at the current state of affairs in research data management.

Taking Dendro as a testbed, a second experiment was carried out at the University of Porto with the collaboration of 11 research groups from distinct domains. This consisted of two rounds of dataset description, carried out with elements of our panel of researchers. From each research group we required the collaboration of three elements (one senior researcher and two other active members of the research team). Each of the two active members was involved in a separate stage of the experiment.

To perform several curatorial tasks that supported the experiment (user interviews, ontology design, configuration of software and metadata quality assessment) we had the collaboration of four researchers from our research group, with backgrounds that included Information Science, Librarianship and Psychology. Preliminary work included meetings with senior researchers to perceive data managements needs and metadata requirements. After these initial contacts, their informal requirements for metadata were formalized into domain-specific ontologies [50, 48].

The goals of our experiment were to prove that:

1. Descriptor selection can be partially automated by the adoption of recommendation techniques that learn from each research group's descriptor usage patterns.

2. Domain-specific metadata can be produced more easily (i.e. with less effort on behalf of the users) through the adoption of the proposed semi-automated descriptor selection techniques, and

3. The proposed approach is generalizable to practically any research domain.

These three goals are related to our proposed research questions as a whole. Involving researchers in the experiment allows us to answer research question 1, which concerns the ability of a collaborative Research Data Management (RDM) platform to engage researchers in the description of their datasets. We also draw conclusions on the ability of a recommender system to assist researchers in the selection of adequate descriptors for their datasets (research question 2), while maintaining the quality of the resulting metadata records (research question 3).

In this chapter we present the context of this experiment, the workflow carried out during our evaluation, and an analysis of results gathered from it.

12.2 EXPERIMENT

Recommender systems define three main categories of evaluation experiments: *offline studies*, where the system is tested against current baselines using relevant datasets (usually very large), *user studies*, where a small set of users is asked to carry out certain tasks and are then asked questions about their experience, and finally *online experiments*, where the system is tested on a large scale, without users knowing that they are participating in an evaluation experiment [203].

Our recommendation task is very specific and there is no reference dataset, nor any past reports of an experiment of this kind, which made us exclude the offline study. The online experiment, on the other hand, is usually the most expensive option, given the need for an operational system, as well as a very large number of users. *Crowdsourcing* can help gather large numbers of test users, but it is only applicable to highly controlled workflows where paid, non-expert workers perform simple, independent *micro-tasks*. Complex, real-world tasks that cannot be decomposed into independent simpler tasks are not appropriate for this method [185]. We ruled out the online experiment because the data description task is complex and not divisible into micro-tasks. Moreover, research dataset description requires the participation of users with substantial scientific knowledge on their respective domains, narrowing down the users that can participate in this study.

Our selected approach, the user study, is a compromise between the offline study and the online experiment. While it still requires a realistic system for users to carry out the tasks, the supervised nature of the study allows us to accompany the users, explain basics of data description and gather their feedback in a more personalised manner. Our particular study, however, retains a very important element usually present only in online experiments: the fact that users were unaware that we were evaluating their

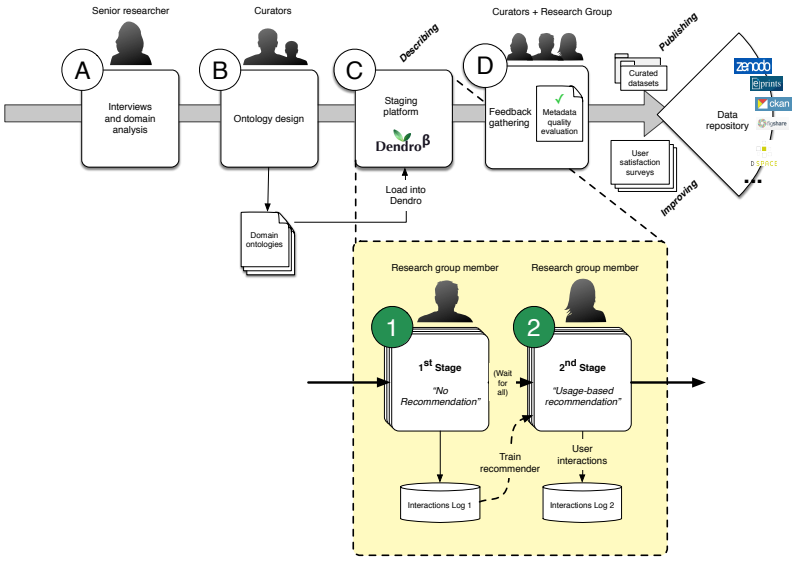


Figure 27: An outline of the experiment

behaviour with respect to metadata descriptors. From their point of view the experiment was a user test of a data management platform.

We have dedicated a Dendro instance to this experiment so that all interactions were associated with it. The instance was prepared and loaded with the ontologies that had been developed for all the domains participating in the experiment. We are focusing here on the activities within the description step of the Dendro workflow, although the experiment also provided interesting data for curators in the research groups, as they reflect on issues of metadata quality.

12.2.1 Participants

The users that participated in the experiment are researchers currently affiliated with 11 research groups in the University of Porto. Previous contacts with people in these research groups had been established, first in the context of a scoping study concerning the status of research data management in U.Porto [186], and then in several initiatives that led us to gather a growing panel of researchers to participate in our research data management initiatives [48, 50]. The research domains of the participating research groups are listed in Table 11.

Overall, our user study was set up according to the regular data management workflow with Dendro. Figure 27 is another view on the Dendro workflow shown in Figure 12, this time with some extra detail on the description step.

The senior researcher represented in **A**, who participates in the gathering of metadata requirements, is involved in the development of the domain ontology, and is left out of this experiment. Two researchers from each group participated in the description (**C**)—one in the first stage and the other in the second stage. These two researchers were themselves senior enough to understand the concerns of research data management (we had no undergraduate students, for example). All were currently involved in

Table 11: Domains of the participating research groups

Overall research area	**Description**	**Example descriptors**
Fracture mechanics	Experimental datasets from fracture mechanics tests over samples of different materials.	Initial crack length and Material type [47].
Biodiversity	Observational datasets for biodiversity. Primary descriptors follow the INSPIRE recommendation [25]	Reference system identifier and Metadata point of contact [191].
Hydrogen generation	Chemical engineering studies on hydrogen generation.	Catalyst, Reagent
Optimisation	Studies regarding algorithms for cutting and packing problems.	Solver configuration, Optimization strategy and Heuristics used
Analytical Chemistry	Chemical engineering experimental data for pollutant analysis.	Analysed substances, Sample count
Social and behavioural sciences	Datasets that result from field campaigns applied to social and behavioural studies.	Methodology, Sample procedure, Kind of data
Computational Fluid Dynamics	Solving fluid dynamics problems using computational methods.	Flow Case, Initial Condition, Temporal Discretization.
Vehicle Simulation	Traffic simulation studies in urban context. Details about the development of this ontology were published [50].	Driving cycle, Vehicle Mass
Oceanographic Biology	Biological oceanography observational and experimental studies on crustaceans [79].	Life stage, Species count, Individuals per species
Solid Earth sciences	Datasets gathered from sensor networks.	No specific descriptors introduced.
Neurological studies	Studies on neural behaviour while performing intellectual tasks.	No specific descriptors introduced.

activities dealing with data collection and analysis, but none of the selected participants had any previous experience with the Dendro platform.

At each stage, and with each group, we carried out a meeting with the researcher and two members of our team. In this meeting, one person provided an overall presentation of the goals and features of Dendro, and proceeded to ask the participant to use the platform and carry out some tasks. The second team member recorded the informal interactions (overall time spent, questions, hesitations, difficulties, suggestions). The meetings were conducted by pairs of elements of the Dendro project team, in different configurations. A total of 5 people participated in the meetings, all knowledgeable in the platform. Meeting were scheduled accounting for a 1-hour period each, but actually took between 1 and 2 hours. Participants were left at ease in the description tasks, so these were only terminated when the participant indicated they had no further contribution.

The meetings took place in the workplace of the participants: a laboratory, meeting room or office where the participants were in their own working environment. The series of meetings in Stage 1 took place in one and a half week, and the same for Stage 2. The meetings provided plenty of feedback concerning the participants relation with their data, views on the research data management activity, and expectations with respect to data publication and sharing. After the sessions, each participant received a short questionnaire. The answers to the questionnaires were collected together with the minutes of the meetings as qualitative data to be analysed in the near future.

The evaluation experiment could have been performed independently for each domain, and in this case the profiles of researchers involved in each experiment would be similar. This was not the option in this study, for several reasons:

1. The main point of Dendro is to provide a common platform for researchers in many areas (the long tail) and not to evolve to specialised disciplinary tools;

2. The tasks required of researchers were not domain-specific, but general enough to be evaluated across domains;

3. One of the expected strengths of Dendro is the possibility of reusing descriptors across disciplines; this requires experiments where a large set of descriptors is provided to researchers, favouring cross-disciplinary use.

Recommendations are derived from the training dataset gathered from the first 11 users, who performed more than 950 interactions. Thus, this user study had 11 expert users producing the training dataset on the first stage and then another 11 users validating the recommendations in the second stage. The number of users typically involved in reported user studies are of this order. Some recent examples involved questionnaires over 34 undergraduate students [117], 15 non-descript users [84], or the analysis of metrics such as precision and recall using the feedback of 17 users [212].

In our case, we submitted questionnaires and analysed several quantitative metrics including the average rank of the selected descriptors in the recommendation list as they were selected, which is a measure of precision.

It would be better to have more users, and we expect to be able to repeat the experiment with a larger sample. Moreover, the experimental data concerning data description tools is scarce, and preliminary experiments, albeit limited, can provide evidence to guide more comprehensive ones.

12.2.2 First stage

The first round of meetings is represented as Stage 1 in Figure 27. In this stage, the Dendro instance was used in its basic configuration, without the recommendation features. In this configuration the descriptors are grouped by ontologies, and these are named after the research domain of the corresponding group. During this stage of the experiment, users could add descriptors to the metadata editor by first selecting the ontology from which they wanted to draw those descriptors, and then selecting them.

Information collected at this stage was used as the baseline for comparison with the recommendation-enabled second stage. Note that this baseline, where the system presents users with alphabetically ordered list of descriptors, is both realistic and reasonably strong. Our participants were not confronted with an unordered or very long list of descriptors all put together. The descriptors are separated by ontology, and each of these has an appropriate name, related to the research domain; this leads researchers to navigate to that ontology. The division of descriptors by originating vocabulary is also common in other repositories. DSpace, for example, allows the configuration of multiple metadata schemas so that descriptors can be drawn from them and used in a metadata record.

An alternative baseline could use the order of the descriptors based on their use or reuse in an existing repository. Such statistics are, in fact, available as part of the studies on the actual usage of descriptors in the Dryad repository, for example [86]. This would, however, introduce a strong bias towards DCMI Metadata Terms (DCTERMS) in detriment of more domain-specific schemas, as it remains the most used metadata schema in existing repositories [151]. Since we wish to reduce bias as much as possible in our baseline, we decided to give all descriptors an equal chance of being discovered in Stage 1.

Note that all domain ontologies were made available in both stages of the experiment to all the researchers—despite the fact that each group worked in a single Dendro project, all the ontologies were available to the users regardless of the project they worked with. Besides providing the domain-specific ontologies developed for the domains, Dendro provides users with some generic ontologies. Among these are DCTERMS (for generic metadata such as Title or Subject), Friend Of A Friend Ontology (FOAF) (for people-related metadata such as a relevant Mailbox or a Depiction of an experimental setup), or the Research ontology (see page 111), a generic research-related ontology that contains descriptors such as Methodology (shown in Figure 22).

During this first stage, user interactions were monitored and recorded by Dendro. The dataset of gathered interactions used as the training data for the recommender used in the second stage will be published to allow its reuse.

12.2.3 Second stage

Stage 2 was started after Stage 1 was completed for all research groups. We trained our ranking algorithm for Stage 2 using the interaction logs from Stage 1. For the sake of the study, the parameters of the ranking functions (shown in Equation 9) were not updated with new data throughout Stage 2, even though we continued to gather it for result analysis.

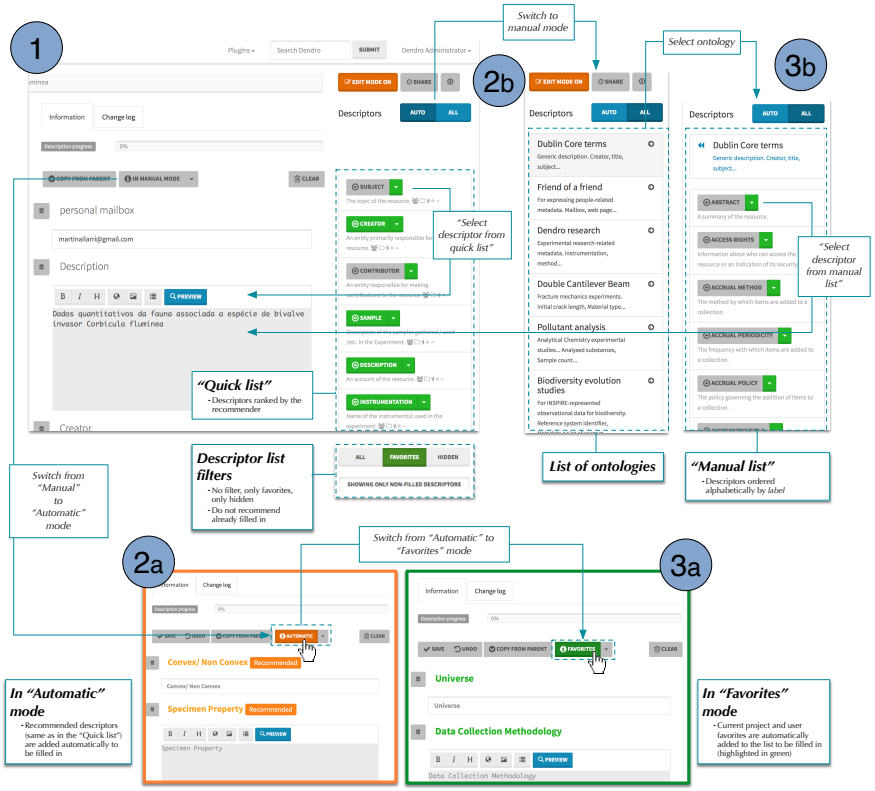


Figure 28: Dendro interface with recommendations enabled

In Stage 2, the descriptor ranking features of Dendro were enabled, and a separate group of participants was engaged in the experiment. The user interactions of this second group of users were again monitored and recorded, in a different log from the one used in Stage 1.

Figure 28 shows the *Recommendation* interface of Dendro. Most file management features remain unchanged, as well as most features of the metadata viewer/editor (**1**). There is, however, the addition of a list of descriptors selected by the recommender. The presentation of every descriptor looks similar to the one in the basic system, except for a small detail: below every descriptor, a series of icons/badges provides hints on the reasons why the descriptor is recommended. Example of these are "Frequently used in project", if it is a "Project Favorite", and other possible features. The goal is to increase transparency, a trait that is beneficial to the effectiveness of recommender systems [207]. Descriptors selected from this list can be added to the metadata editor to be filled in, like they were in Stage 1, and additional types of interactions were recorded due to the presence of the recommendation elements, such as "Select a descriptor from the list of recommendations" (see Table 8).

During Stage 2 users could still select descriptors manually by pressing the All button (**2b**). After changing to the manual mode, they are presented with the same list of ontologies available in Stage 1, and if they select one, the corresponding descriptors are presented in alphabetical order (**3b**). The selection of the ontology counts as an effort interaction, as well as the selection of the descriptor from the alphabetically-ordered list, which is counted

as an interaction of the type "Select a descriptor from the manual list of descriptors".

The recommendation-enabled interface includes a button in the metadata editor that allows users to automatically add all the recommended descriptors to the metadata editor, instead of selecting them one by one from the list of recommendations. This metadata editor mode is called *Automatic mode* (**2a**). By clicking the button a second time, users can switch to a third mode, *Favorites* (**3a**), which automatically adds all descriptors currently marked as favorites (either *Project Favorite* or *User Favorite*) to the metadata editor to be filled in. Clicking a third time on the button will turn off the automatic addition of descriptors, returning it to Manual mode.

12.2.4 Experiment control

The conditions assumed for the experiment, to ensure that all participants had a similar view on Dendro, give rise to a behaviour which is more static than expected in a production environment. The recommender may therefore produce some unexpected results. For example, marking a descriptor as favorite during Stage 2 did not alter its rank. This was necessary to ensure uniform conditions for all users participating in Stage 2. This is not entirely unrealistic, even in a production scenario, as these issues can be considered intrinsic to the update behaviour of the recommender. It is well known that the training of recommendation engines is often a periodic background job executing during the times of lower server loads. As a consequence, recommendations may not be influenced by the immediate actions of the user, even in production systems.

The behaviour of the recommender, although present, was not explicitly noticed by users. From our monitoring of their behaviour during the sessions in Stage 2, we observed that users actually kept favouriting those descriptors that they found most interesting for describing their datasets, instead of directly trying to influence the recommendations provided by the system.

Stage D, the final stage of the Dendro workflow, focuses on ensuring the quality of the metadata records produced through stages A, B and C. In the evaluation experiment, user satisfaction feedback was gathered, with a short anonymous survey being sent to all the participants via email. There were two versions of the survey, one for the users that were engaged in Stage 1 and another for Stage 2 participants. The latter contained the same questions as the former, as well as some additional ones that specifically targeted the effectiveness, relevance and perceived usefulness of the recommendation features that were available. Overall replies were positive, with scores sitting close to 4 in a scale from 1 to 5, and all of them above the value of 3.

Stage D also includes a data publication stage, for those research groups that wish to see their datasets published in a repository. Researchers pointed out some of the main repositories and data authorities in their respective areas of research, and asked if the platform could facilitate the deposit of their datasets in these platforms. After mentioning the sharing capabilities of Dendro, many researchers reacted positively and were encouraged to further describe their datasets. This gave the experiment a realistic nature up to the results for the researchers, which are not part of the evaluation. It has been observed that the investment placed by the researchers in produc-

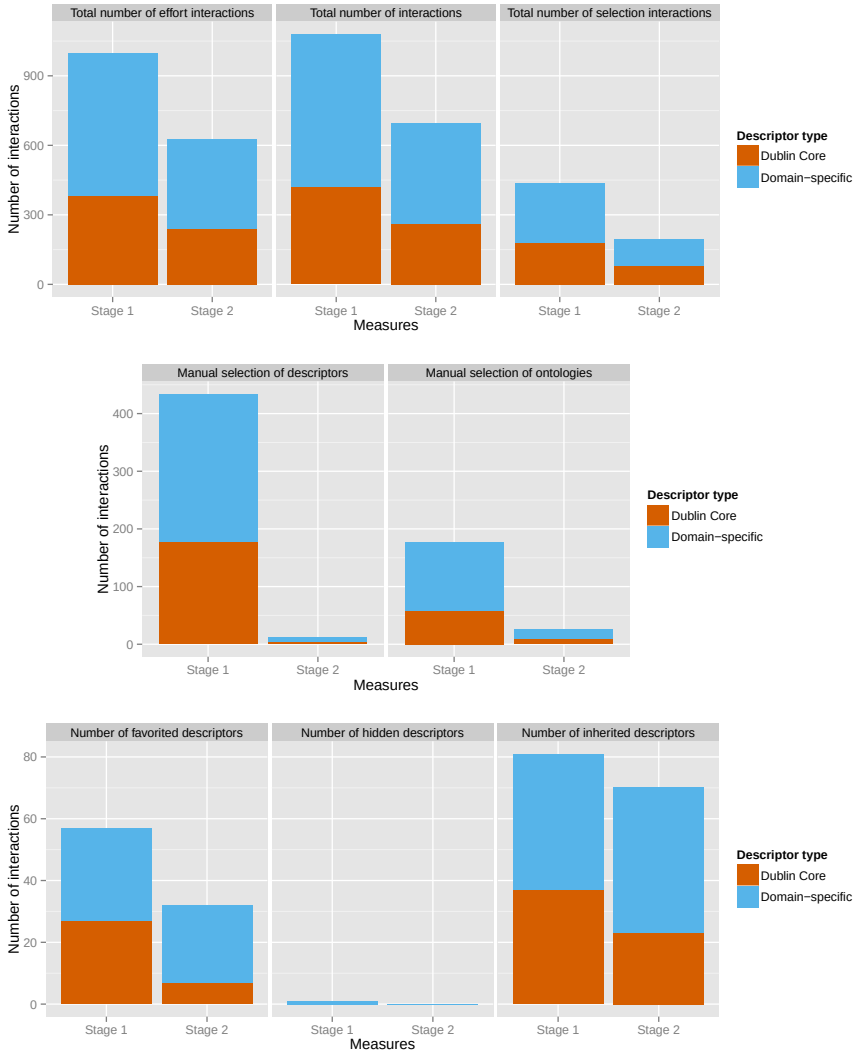


Figure 29: Quantitative analysis of interaction signals

ing metadata for publication is usually much higher than that employed in personal annotations [36].

12.3 EVALUATION RESULTS

In the two stages of the experiment, the participants used Dendro in the process of describing a dataset, and selected descriptors which we now split between Dublin Core and domain-specific. We designate domain-specific descriptors those which are designed around a specific description domain, such as agent or entity description (e.g. the FOAF ontology), research-related metadata, or a research domain. Our intention is to highlight the usage of non-Dublin Core descriptors throughout the experiment, pointing out how Dendro can facilitate the use of domain-specific descriptors by users without data management training. Figure 29 shows an analysis of the recorded interactions in the system, during both phases of the experiment. In these charts, the *Stage 1* columns correspond to the interactions

recorded during Stage 1 of the experiment, and the *Stage 2* columns are relative to Stage 2 of the experiment (see Figure 27). For each bar, there are two stacked columns: Dublin Core and domain-specific. The Dublin Core column indicates interactions performed over descriptors from the Dublin Core Terms metadata schema, and the Domain-specific column indicates interactions over all other descriptors.

We can draw several conclusions from the analysis of these charts. When comparing the number of *manual selections* of descriptors between Stage 1 and Stage 2, we can observe that there were almost no manual selections in Stage 2. A manual selection occurs when a user selects a descriptor from the list of alphabetically-ordered descriptors. The near absence of manual selections in Stage 2 shows that users did not feel the need to select any descriptors from the alphabetical list, since the list of recommended descriptors was good enough for describing their data. This is further confirmed when we compare the number of manual selections of ontologies. In this feature are included those interactions recorded when the user clicks over an element in the alphabetically ordered list of ontologies for the different research domains. Users must navigate between ontologies to add the descriptors that they want to the metadata editor. For example, if the user selects the Dublin Core ontology, then adds 3 descriptors to the metadata editor, then goes back to the ontology list, selects the Vehicle Simulation ontology and adds another two descriptors from that ontology, there are two *manual selection of ontology* interactions: one counted as Dublin Core and another counted as domain-specific. The alphabetically-ordered list of ontologies was also present in Stage 2, but it was much less used, as can be seen in the chart. The difference can be attributed to the fact that users of Stage 2 used the list of recommended descriptors and did not need to search for them in the alphabetically-ordered lists. There was still some manual selection of ontologies, maybe to explore the lists of descriptors that were present in each ontology; given the extremely low number of manual selections in Stage 2, users would only look into the contents of the lists, but would not actually select any descriptors.

The number of descriptor favouriting interactions performed in Stage 2 was inferior to that of Stage 1. In our view, this can be attributed to two reasons. The first possible reason is that, in Stage 2, descriptors were already good enough in the recommended descriptors list, so users would not feel so much the need to improve the recommendations provided by the system. On the other hand, this could also be partially attributed to an experiment design decision: since the recommender for Stage 2 was trained from the logs gathered in Stage 1, users would not see any change in recommendations even after they promoted a descriptor to Project or User Favorite—this is because the favouriting interaction is recorded in a separate log from the one used to train the recommender. In a production-running Dendro, this would seem like a wrong behaviour, but it was necessary to ensure that all the users used the same recommender in the exact same training state throughout the entire experiment.

The number of *effort interactions* represents the number of interactions that required direct interaction on behalf of the user to be performed, and we can see that it was clearly lower on Stage 2 than in Stage 1. This shows that the presence of the recommendations made it easier for users to fill in a similar number of metadata descriptors with a reduced effort. Hiding interactions, which explicitly constitutes a statement of non-relevance towards a descriptor, were inexistent in Stage 2 and insignificant in Stage 1. We can say that

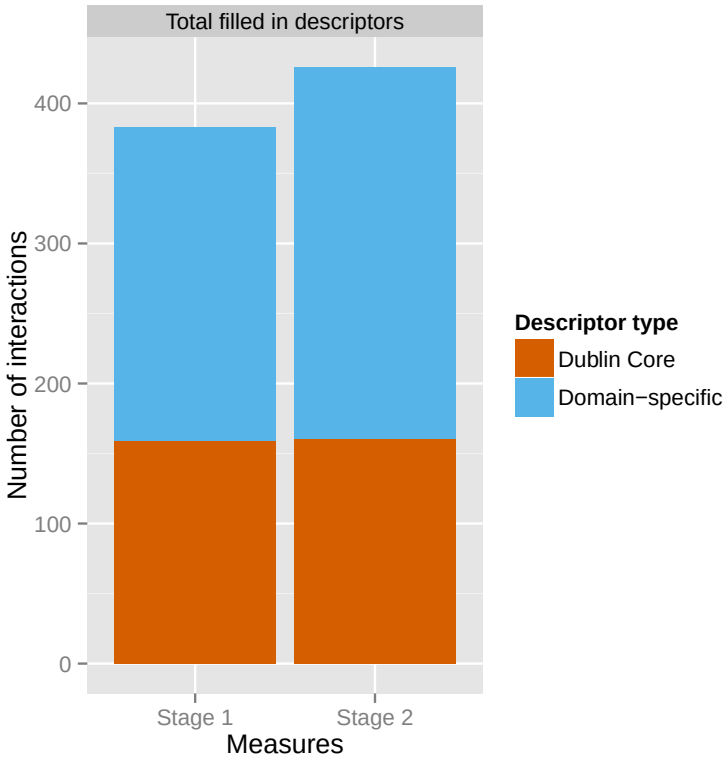


Figure 30: Number of filled-in descriptors

the users had a limited time working with the platform and tended to take a positive approach to the system: instead of hiding all the descriptors they did not like, they simply searched for those that they needed. Otherwise, they would have to hide a lot of descriptors before starting to perform the description work (as there are a lot of possible choices). Since the descriptors were consistently accepted across all domains, there was practically no need to explicitly reject descriptors. Despite not being used very often, we believe that the hiding feature can provide valuable feedback in identifying malfunctions in the recommender, while letting users control the recommendations.

Figure 30 shows the number of interactions that included filling in a descriptor, in both cases. For each of these interactions, a descriptor was actually saved into the system, and thus they are crucial for the evaluation of system performance. We can see that the number of filled-in descriptors does not change much from one scenario to the other, but the Stage 2 scenario still manages to achieve, not only a higher number of filled-in descriptors overall but, more importantly, more domain-specific descriptors, which was one of our goals.

This result, combined with the ones regarding the number of effort interactions and the number of overall interactions, allows us to conclude that with the recommendation system in place (Stage 2), users were able to fill even more descriptors, with the added advantage of much less interactions that required effort from them.

Another interesting metric is the position of the descriptors when they are selected. When users do not find the descriptors that they need in the top of the descriptor list, they will scroll further down until they do (or until they

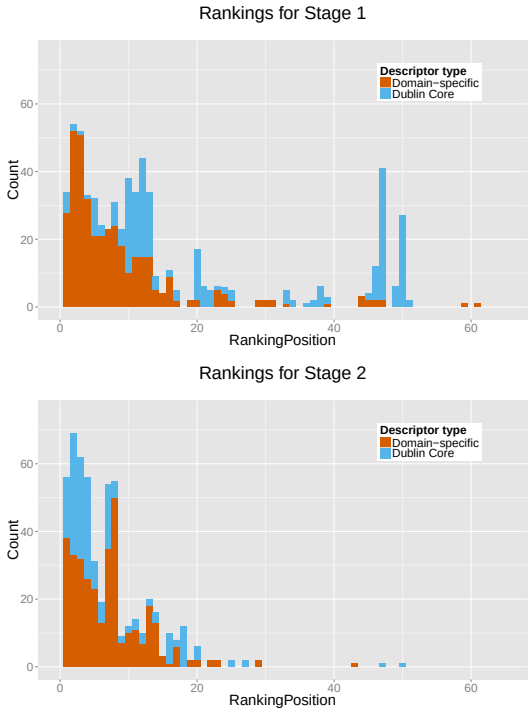


Figure 31: Histograms of the rank of the selected descriptors

give up). These scrolling actions are undesirable because they indicate poor quality of the ranking. Figure 31 compares the distribution of the selected descriptors's rank between Stage 1 and Stage 2. On the horizontal axis of each chart is the rank (position in the list) of the descriptor at the time of the interaction—-lower values indicate a position nearer the top of the list—and on the vertical axis there is the absolute frequency of interactions performed over descriptors placed on that position. We can see a clear difference in the distributions.

During Stage 1, when the descriptors were ordered alphabetically, there is a broader distribution of the user interactions across the ranks, meaning that users interacted over descriptors that were further down the list. During Stage 2, there is a clear concentration of interactions in the top 20 positions of the ranking. This result indicates a clear improvement in Stage 2 over Stage 1, as it shows that the recommendations saved users the effort of scrolling up and down to find relevant descriptors.

12.3.1 Description learning curve in Dendro

The observation of the overall user behaviour provides some insight on the response of the users to the system. The inclusion of the recommendation features in the Dendro platform produced a change in how users learned to use the platform. While in Stage 1 users tended to browse around the list of ontologies to get a feel for where the descriptors of different types were located, Stage 2 users moved much faster to selecting descriptors and filling them in. The reason for this is that they immediately recognised the recommended descriptors as being relevant to their research domain, hence they knew instinctively what to do as they entered the platform. We could therefore conclude, through observation, that the recommendation

introduced in Stage 2 positively influenced Dendro's learning curve, helping researchers get acquainted with the platform faster than those in Stage 1.

We also conjectured that Stage 2 users might be tempted to analyse how their Stage 1 colleagues performed their descriptions, since Stage 2 users worked in the same project as those in Stage 1. We were pleased to observe, however, that Stage 2 users quickly felt comfortable enough to start describing their own datasets using the recommended descriptors instead of first trying to figure out how their colleagues performed their task.

12.4 CONCLUSIONS

There is large amount of work to be done on research data management and many problems to be solved on metadata creation, especially in a multidisciplinary context. In this chapter, we evaluated our proposal for an agile, usage-driven solution for the selection of appropriate descriptors, targeted at the production of metadata records for research datasets.

While the size of our test group was small when compared to many recommender systems evaluation tasks, we must highlight the fact that it was carried out with expert users in a realistic scenario, and such experiments are not common because they incur a high cost in time and development. Another challenge that we had to overcome was the lack of a baseline, as is the norm in recommender system studies. In our case, due to the novelty of the task and its specialised nature, there is currently no baseline. This led us to divide our set of users into two groups, so that we could first establish a baseline (Stage 1 or our experiment) and then build our recommender to demonstrate its better performance when compared to the baseline (Stage 2). The size of the user group is similar to those found in other user studies carried out to evaluate recommender systems.

Our results show that an usage-based descriptor selection approach can greatly reduce the effort of data description across different research domains without compromising on the quality of the metadata records. The interaction logs gathered during Stage 2 also express a very strong preference for the recommended descriptors in detriment of manual selection. We have proved the value of this approach towards the gathering of more comprehensive metadata, which is vital for the successful publishing and reuse of research datasets.

An unexpected but positive side effect was that there was an easier adaptation to Dendro when the recommender was present. During Stage 2, users that had no prior contact with the platform would start to describe their data faster and with less doubts about the features of the system than those that participated in Stage 1.

The data analysis presented here is preliminary, but results are promising in what concerns metadata gathering in research contexts. However, even with the data already collected, there are further analyses to explore, both with the quantitative data logs and with the qualitative meeting minutes.

Part V

Conclusions

13 DISCUSSION

In contrast with research publications which can be described by curators long after projects are finished, datasets need to be described in close collaboration with the researchers that participate in their creation to allow their review and reuse.

While a publication is usually described reasonably well using generic metadata, research datasets need more detailed and domain-specific metadata to allow their *scientific review*. To gather this domain-specific metadata, domain creators should be involved in the description process, thus allowing their peers to interpret the data and possibly reuse it. As we discuss in Chapter 2, the challenge lies in the fact that they are not data management experts and can sometimes regard metadata production as yet another task to add to a myriad of others, but without apparent benefits. Descriptors also vary greatly from research domain to research domain and can be overwhelmingly numerous. To assist these non-expert users in the task of metadata production without disturbing their day-to-day work, we decided to study if we could use interaction logs in a recommendation system to help them deal with this *information overload*. Our hypothesis summarizes our goal: it is possible to use past usage information to recommend both generic and domain-specific descriptors—and as such, greatly reduce the effort required to produce high-quality metadata records?

Thesis Statement. *Usage information can be used to build application profiles for the description of research data assets, in a semi-automated manner, reducing the effort required to produce high-quality, domain-specific metadata records.*

Through our user study, we have determined this hypothesis to be true.

13.1 ENGAGING RESEARCHERS IN THE MANAGEMENT OF THEIR DATASETS

The data description process must start very early in the research workflow; this agility may seem like a detail at first, but it makes all the difference. We aim for researchers to have their base data almost completely described by the time their research is reported, even allowing them to cite their published datasets at that point. Missing this opportunity for data publishing is often unrecoverable, as researchers have to engage in new projects. If the process is delayed, data creators may be already working on different institutions or projects, and the original data gathering contexts may be forgotten, at least partially. This led us to formulate research questions 1 and 3.

Research Question 1. *Can a collaborative research data management environment engage researchers in the management of their own research data?*

Research Question 3. *Can researchers, when supported by collaborative research data management tools, produce metadata records that are satisfactory as judged by a curator?*

We have answered these questions through our user study (Chapter 12), where researchers described their own datasets using Dendro, our prototype Research Data Management (RDM) platform, which included a ranking-based descriptor recommender. The resulting metadata was domain-specific, complete and high-quality as judged by a curator as well as elements of the research groups. The results of our questionnaires about the usage of the Dendro platform also indicated that it was met with very positive responses by the participating researchers.

13.2 SUPPORTING OUR WORK ON THE FEEDBACK OF RESEARCHERS

All stakeholders are relevant in the research workflow, but researchers are perhaps the most important. Realizing this, we have continuously supported our work on their requirements, asking for their advice when designing the workflow that we propose for research data management—outlined in Section 7.3.

The requirements for Dendro (outlined in Chapter 7) were drawn from meetings with researchers, the lightweight ontology modeling process (proposed in Chapter 9) is based on researcher input, and our evaluation user study (described in Chapter 12) was carried out with the collaboration of 11 research groups.

13.3 USING TRIPLE STORES AS A MORE FLEXIBLE DATA REPRESENTATION

As we have determined in our analysis of the relational schemas of several data and publication repositories (Section 7.4), their data models are not designed to handle flexible metadata whose descriptors can be modified as time passes.

In contrast, our approach for the design of Dendro's database allows curators to grow the data model without changing the platform's code base. At the same time, it deals with three requirements which are hard to implement in a relational model: allowing users to use a variable number of descriptors, which are unknown at the time of system design and implementation, whose values can be modified over time, and which need to be applied to structures of files and folders. To the best of our knowledge, no other solution so far has applied a similar approach to this problem.

13.4 USING INTERACTION INFORMATION AS EVIDENCE FOR DESCRIPTOR RECOMMENDATION

As we have shown, interaction information can be a good enough source of data for a recommender to quickly provide good descriptor recommendations in an unobtrusive way. This means gathering little to no feedback from the user and interfering as little as possible with the description workflow, while still providing relevant descriptor recommendations.

Users' response to the recommendations was overwhelmingly positive, as shown by the reduced number of manual descriptor searches and selections after the recommendation system was put in place. Our results also show that an usage-based descriptor selection approach can greatly reducing the effort of data description across different research domains, answering research question 2.

Research Question 2. *Can usage information be used to provide adequate recommendations of descriptors for distinct research domains?*

13.5 NOVELTY, POTENTIAL FOR IMPROVEMENT AND FUTURE STEPS

Our work is novel as it can be seen as a first baseline for future works towards metadata generated and driven by research groups and communities. The recommender that we built can certainly be improved, the evaluation step can be made stronger with a larger number of test users or a full-fledged online experiment, and the data analysis presented here may be improved; however, results are promising in what concerns metadata gathering in research contexts.

As we see in most data management platforms, overviewed in Chapter 3, strict adherence to a single schema or Application Profile (AP) is commonplace. The implications of this work for the design of a data management platform are also relevant. It is a first evidence that perhaps system designers should consider diverse metadata descriptors and allow the community to naturally build their own APs, instead of picking a single metadata schema or AP for them.

The modeling of descriptor ontologies was an expensive process because it required the collaboration of researchers through several meeting where their description needs would be gathered and formalized. However, as more and more research domains are covered by this approach, less modeling of new descriptors is likely to be required when describing datasets, thus relying less on the manual work of curators.

Further analysis will be performed as we continue research on this topic, but so far we have proved the value of this approach towards the gathering of more comprehensive metadata, which is vital for the successful publishing and reuse of research datasets. More concrete research ideas that stemmed from this work will be presented in Chapter 14.

13.6 RESEARCH CONTRIBUTIONS TIGHTLY RELATED TO THIS WORK

This research could not have been carried out without the vital contributions of other researchers in InfoLab, whose multi-disciplinary skills allowed Dendro and its workflow to be implemented as it is.

In particular, we highlight the contributions of João Aguiar Castro on the usage analysis of prototype data management tools at U.Porto [49] and lightweight ontology design integrated in the Dendro platform [48, 50, 191]. We also highlight Ricardo Carvalho Amorim's overview on research data management solutions [13] as well as his excellent work on LabTablet, an

electronic laboratory notebook with full integration with Dendro [10, 11, 12]. Both worked tirelessly in meetings with researchers, helping to test Dendro, accompanying the meetings with them and providing guidance and support whenever necessary—not only to the researchers but also to other junior researchers in InfoLab which also participated in setting up the experiments.

14 FUTURE WORK

Regarding the technical aspects of Dendro, we propose improvements over the recommendation algorithm, new evaluation scenarios, and possibilities for making it more standards-compliant. As for future research, we are planning to use Dendro to attest the benefits of a collaborative data management workflow at U.Porto. Among our future research lines are a distributed environment supported by Dendro for sharing not only high-quality metadata but also datasets, and the possibility of sharing generated APs as standalone ontologies to foster reuse and improvement. A combination of these two (serialization and sharing over a distributed network) would provide a fast-paced environment for AP review and improvement, driven by real usage in distinct research domains.

14.1 FURTHER ANALYSIS OF THE GATHERED DATA

Concerning the recommendation approach proposed here, there are many aspects we had to simplify to be able to get conclusions and many more to be experimented in future studies. However, even with the data that we collected during this user study, there are further analyses to explore, both with the quantitative data logs and with the qualitative meeting minutes.

This kind of user experiments are expensive in time, require the willingness of the participants and in some cases need new participants at each round, so a lot of commonsense is required when formulating hypothesis which need to be validated via experiments with subjects.

14.2 IMPROVEMENTS TO THE RECOMMENDER ALGORITHM

Content-based recommendation could be used to match descriptors to the profiles of users or to Dendro projects. In fact, a previous version of our recommender implemented a content-based approach, with descriptors being selected based on their source ontology. This way, if a user used a descriptor from an ontology, the system would recommend more descriptors from the same ontology. In the version of the recommender used here, those features were not present, however, and therefore were not studied. This is another possibility for future improvement of the recommendation engine; more features are of course likely candidates for inclusion, and we will continue to study such alternatives.

Dendro already performs textual indexing to enable free-text search based on the text content of files. In the future, we can use these texts to introduce a content-based recommendation component into the existing ranking approach. A possible way to do this would be to analyse the rdf:comment property of every descriptor and then to associate the most important terms

(calculated through a $tf \cdot idf$ measure, for example) to the important terms of the project's dcterms:description or a file or folder's dcterms:description.

We already consider the topology of the graph within each Dendro instance when building the descriptor ranking: all files and folders belonging to a Dendro project are connected by links, and those links are used to determine all the interactions that were performed over files and folders belonging to a project. However, link analysis algorithms could take this a step further in the future, in order to improve the performance of this recommender. For example, if our users specify links between files and folders through certain descriptors such as dcterms:references or dcterms:isReferencedBy, those links and folders could be taken as the edges and vertexes of a graph to be analysed through a PageRank or HITS algorithm. After calculating the scores of the files and folders, their values can be used as weights for the importance of the descriptors that link those folders. A different ranking will then emerge, based not only on interactions (as we did in this work) but on the topology of the graph that links the metadata records.

14.3 DIFFERENT EXPERIMENTAL SCENARIOS

In the experiment documented in Chapter 12, the ranking algorithm was configured as a static version, and the interactions of Stage 2 were not used to retrain the system. As future research, we would like to study how users respond to a scenario where they are presented with dynamic recommendations, which rapidly change depending on their actions in the system.

14.4 DESCRIPTOR SELECTION OUTSIDE OF A DATASET DESCRIPTION CONTEXT

Past works have asked users were prompted to select, from three ranked lists, the one that they thought contained the highest number of relevant items [212]. The first goal was to establish that there was an increase in accuracy when results from an *implicit feedback*-based recommender were presented side by side with a ranked produced by purely random recommendation. To further assess this, two of the three ranked lists were fed by random recommendations, and a third was fed by the proposed recommender system. Users consistently noted that the rankings yielded by the recommender system contained a higher number of relevant items than the rankings that were filled with random recommendations.

Figure 32 shows a mockup of a feedback gathering system for the evaluation of recommendations in a hypothetical experiment along the same lines. On the left and middle ranked lists of descriptors, their selection and ranking is calculated randomly. In the rightmost ranking, the descriptors are produced by the Dendro recommender. Researchers would be asked to select, from the three lists, the one that contained the set of descriptors that they would likely use in the description of a dataset from their particular domain. The process would be repeated a number of times, and for each time a new set of three ranked lists would be presented to the user. On each time the lists were presented, their position would also be randomized. While the source of the descriptors (Random or Recommender-based) is clearly visible in the picture, the lists presented to the users would not

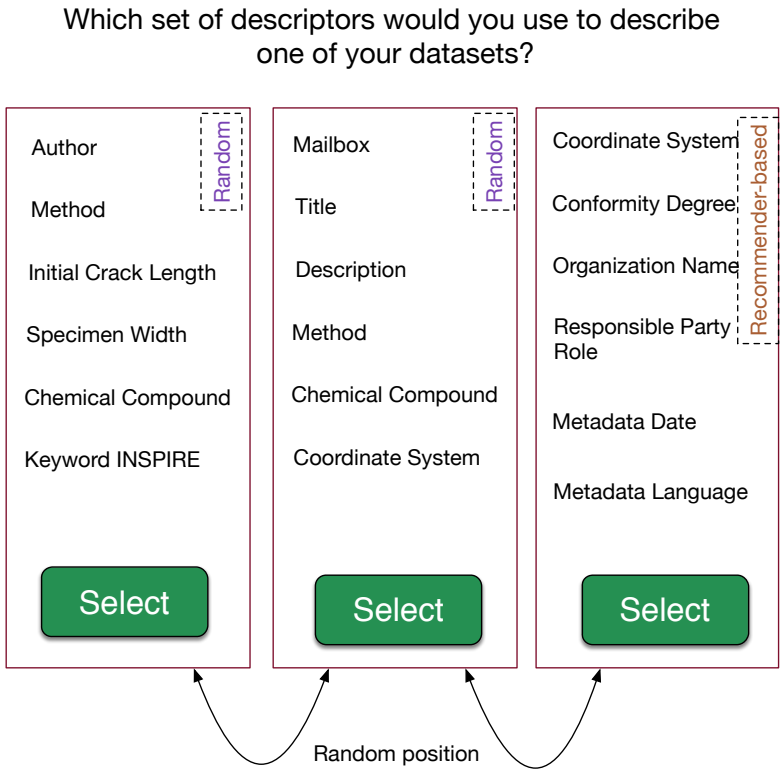


Figure 32: Mockup of a prototype evaluation interface for a future evaluation experiment

show such distinctions, and all lists would be visually identical in terms of design elements.

14.4.1 Learning to rank

In this experiment, the $f_{i_{max}}$ and r_i parameters of the feature scoring functions were set ad-hoc from our intuition on the domain. We opted for this approach because we knew that the number of users would be insufficient to provide enough interactions for a *learning to rank* approach.

Learning to rank refers to the application of machine learning techniques to train the model used in a ranking task [132]. For our recommendation task, a learning to rank approach could be used to make $f_{i_{max}}$ and $f_{i_{min}}$ vary depending on the response of the users to the lists of recommended descriptors. In time and with a large enough number of interactions, we would expect them to converge to the optimal values in the context of a user or a project. Since the primary focus of this work is not on the ranking algorithm itself but on its application, we leave that research line open for a future work.

14.5 FORMALISING APPLICATION PROFILES AS ONTOLOGIES

Dendro could be complemented with a module for serialising an APs as ontologies, including the descriptors that are included in the AP, any validation rules for the values introduced in each field, provenance information for the AP, or other relevant information. The introduction of this serialization and deserialization module could foster the improvement of such profiles by allowing different Dendro instances to reuse APs according to the domains of the researchers that use it. Any reuse information on these APs could also prove an additional source of evidence for determining the most used descriptors and APs.

14.6 FEDERATED DATA MANAGEMENT USING DENDRO

The Dendro platform was designed to be scalable in an institutional infrastructure and easily integrated with existing solutions, hence the emphasis on building a well structured Application Programming Interface (API) for all the services provided by the platform. This interoperability can be harnessed in the future to bring Dendro instances together in a peer-to-peer network. This would allow for distributed querying, enabling users to search datasets beyond the scope of their Dendro installation, or to implement distributed file backups, in a manner similar to Lots of Copies Keep Stuff Safe (LOCKSS)—perhaps even reusing that solution, as it is open-source software[1].

After an AP serialization module is added to Dendro, even APs could be retrieved from the network, either via suggestions or text query. The desired AP would be fetched from one Dendro instance to another and applied to the current project.

14.7 METADATA QUALITY IMPROVEMENTS

Dendro does not currently validate the formats of metadata values introduced by the users. The controls change from a text box when the descriptor is a text field to a calendar control when the descriptor is a date, but there is no validation on the actual format of the texts. For example, a research group may want its members to write the value of the dcterms:author of a file or folder in the formats FirstName Surname or Surname, Firstname. The DCMI Metadata Terms (DCTERMS) schema does not specify such rules; however, they could be specified in the AP adopted by the particular research group, as a regular expression or a short piece of code to validate each field when saving the metadata record. Another option would be to introduce Web Ontology Language (OWL) restrictions on our lightweight ontologies, but that would make them harder to model; a compromise between operational and modeling detail needs to be established.

1 See http://www.lockss.org/support/open-source-license/

14.8 COMBINING DATA WITH METADATA

So far, Dendro has been designed to represent only metadata records as triples, not performing any data conversion or extraction operations over the deposited files. In the future, we consider the implementation of such features, in order to allow data subsetting, referencing and exploration, as it has been requested by some of the researchers in our panel. VoID, the Vocabulary for Interlinked Datasets [7] offers an interesting representation for the relationships between data subsets.

14.9 COMPLIANCE WITH STANDARDS

We have included in the development of Dendro the implementation of modules for the integration with external repositories. They were designed according to the APIs, and we have yet to comply with existing recommendations in the structure of our exported ZIP files. We thought of them as a means of personal dataset backup for researchers instead of exporting functionalities *per se*, so their structure is identical to that presented in the Dendro File Explorer (see Figure 17). In the future, we plan to make their structure compliant with existing specifications, such as the BagIt Packaging Format [38] or the Research Object Bundle [209].

14.10 PUBLICATION OF DATASETS, ONTOLOGIES AND BASE DATA

The principle in Dendro that the descriptions and files are not locked inside a prototype platform gives researchers additional assurance that their effort has a concrete and useful purpose and contributed to the realism of the experiment presented here. We are now working on the publication of the datasets deposited in the course of the experiment, as we get the final approval from the members of our research panel. The data collected in this experiment will also available soon in a citable record, so that others may replicate our analysis or explore it further. For now, we are making it available in a demo instance of Dendro[2]. The developed ontologies are also available in that same demo instance and in EUDAT[3].

14.11 WRAP-UP

This work demonstrates how users with little data management background but with expert knowledge in their research domains can produce high-quality, domain-specific metadata records, as long as they are supported by an adequate, user-friendly RDM platform. Dendro is an example of such a platform, as it can help them select the metadata descriptors that suit their research domain and their datasets. For this, the platform uses a recommender system trained by descriptor usage and other user interaction data.

2 http://dendro.fe.up.pt/demo
3 https://b2share.eudat.eu/record/292

Descriptor usage patterns can be used to recommend descriptors and help researchers, who are not data management experts, produce good metadata records themselves. Given the amount of research data in production and its diversity, such an approach may be not only beneficial but necessary as more and more datasets enter RDM workflows.

Such approaches might also assume an important role in metadata standard design processes in the future. By sharing descriptor usage data and statistics, researchers and communities in the long-tail of science can contribute towards the faster design, improvement and adoption of standard application profiles for their domains. This would complement the current approaches, which are mainly driven by broad agreements between curators, data management professionals and researchers with data management skills.

This work will be further explored as we expand our researcher panel to more research groups from the University of Porto and continue to improve Dendro as well as the recommender.

BIBLIOGRAPHY

[1] G. Adomavicius, N. Manouselis, and Y. Kwon. «Multi-criteria recommender systems». In: *Recommender Systems Handbook*. 2nd. New York, NY, USA: Springer-Verlag New York, Inc., 2010, pp. 769–803. ISBN: 9780387858197. DOI: 10.1007/978-0-387-85820-3{_}24 (cit. on p. 63).

[2] E. Agichtein, E. Brill, and S. Dumais. «Improving web search ranking by incorporating user behavior information». In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), pp. 19–26. ISSN: 08963207. DOI: 10.1145/1148170.1148177 (cit. on p. 135).

[3] H. S. Al-Khalifa and H. C. Davis. «The evolution of metadata from standards to semantics in E-learning applications». In: *Proceedings of the seventeenth conference on Hypertext and hypermedia - HYPERTEXT '06* (2006), p. 69. DOI: 10.1145/1149941.1149956 (cit. on p. 17).

[4] F. et. Al. *Content Negotiation, part of Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616)*. URL: http://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html (cit. on p. xvii).

[5] A. W. Alam and N. Schumann. «datorium: Sharing Platform for Social Science Data». In: *Proceedings of the 14th International Symposium on Information Science (ISI 2015)*. May 2015. 2015, pp. 244–249 (cit. on p. 43).

[6] G. Albuquerque, T. Lowe, and M. Magnor. «Synthetic generation of high-dimensional datasets». In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2317–2324. ISSN: 10772626. DOI: 10.1109/TVCG.2011.237 (cit. on p. 71).

[7] K. Alexander, R. Cyganiak, M. Haysenblas, and J. Zhao. «Describing linked dataset - On the design and usage of VoID, the "Vocabulary Of Interlinked Datasets"». In: *Proceedings of the Linked Data on the Web Workshop (LDOW 09)*. Madrid, Spain, 2009 (cit. on pp. 21, 111, 169).

[8] R. Alhajj. «Documenting Legacy Relational Databases». In: *Lecture Notes in Computer Science* 1727 (1999), pp. 161–172 (cit. on p. 80).

[9] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. «Basic local alignment search tool.» In: *Journal of molecular biology* 215.3 (1990), pp. 403–410. ISSN: 0022-2836 (Print). DOI: 10.1016/S0022-2836(05)80360-2 (cit. on p. xvii).

[10] R. Amorim, J. Castro, J. Rocha, and C. Ribeiro. «LabTablet: semantic metadata collection on a multi-domain laboratory notebook». In: *Proceedings of the 8th Metadata and Semantics Research Conference (MTSR 2014)*. Springer, 2014 (cit. on pp. 6, 96, 164).

[11] R. Amorim. «LabTablet: A multi domain laboratory book». Masters' Thesis. Faculdade de Engenharia da Universidade do Porto, 2014. URL: https://repositorio-aberto.up.pt/handle/10216/73291 (cit. on pp. 8, 164).

[12] R. Amorim, J. A. Castro, J. Rocha, and C. Ribeiro. «Engaging researchers in data management with LabTablet, an electronic laboratory notebook». In: *Proceedings of the Symposium on Languages, Applications and Technologies (SLATE '15)*. Ed. by J.-L. Sierra-Rodríguez, J. P. Leal, and A. Simões. 2015, pp. 111–116. ISBN: 978-84-606-8762-7 (cit. on p. 164).

[13] R. Amorim, J. Castro, J. Rocha, and C. Ribeiro. «A Comparative Study of Platforms for Research Data Management: Interoperability, Metadata Capabilities». In: *New Contributions in Information Systems and Technologies* 1 (2015), pp. 101–111. DOI: 10.1007/978-3-319-16486-1 (cit. on pp. 27, 77, 96, 163).

[14] M. Annamalai and L. Sterling. *Guidelines for constructing reusable domain ontologies*. 2003. URL: http://aisl.umbc.edu/resources/181.pdf{\#}page=74 (cit. on p. xviii).

[15] C. Armbruster and L. Romary. «Comparing repository types: challenges and barriers for subject-based repositories, research repositories, national repository systems and institutional repositories in Serving Scholarly Communication». In: *International Journal of Digital Library Systems* 1.4 (2009). DOI: 10.4018/jdls.2010100104 (cit. on p. 25).

[16] M. Assante, L. Candela, D. Castelli, P. Manghi, and P. Pagano. «Science 2.0 Repositories: Time for a Change in Scholarly Communication». In: *D-Lib Magazine* 21.1/2 (2015) (cit. on p. 44).

[17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. «DBpedia: A Nucleus for a Web of Open Data». In: *ISWC'07/ASWC'07 Proceedings of the 6th International Semantic Web Conference and 2nd Asian Conference on Asian Semantic Web*. ISWC'07/ASWC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 722–735. ISBN: 3-540-76297-3, 978-3-540-76297-3 (cit. on p. 81).

[18] M. Baca. «Practical Issues in Applying Metadata Schemas and Controlled Vocabularies to Cultural Heritage». In: *Cataloging & Classification Quarterly* April 2014 (2003), pp. 37–41. DOI: 10.1300/J104v36n03 (cit. on pp. 14, 109).

[19] T. Baker. *DCMI Usage Board Review of Application Profiles*. 2003. URL: http://dublincore.org/usage/documents/2003/02/11/profiles/ (cit. on p. 18).

[20] Balabanović, Marko and Y. Shoham. «Fab: content-based, collaborative recommendation.» In: *Communications of the ACM* 40.3 (1997), pp. 66–72. ISSN: 00010782. DOI: 10.1145/245108.245124 (cit. on pp. 58, 64).

[21] A. Ball. *Scientific data application profile scoping study report*. Tech. rep. June. Bath, UK: UKOLN, University of Bath, 2009. URL: http://homes.ukoln.ac.uk/{~}ab318/docs/ball2009sda/ (cit. on pp. 12, 77, 78, 125).

[22] A. Ball. *Tools for research data management (version 1.0)*. Tech. rep. Bath, UK: University of Bath, Innovative Design and Manufactoring Research Centre, UKOLN, DCC, 2012. URL: http://opus.bath.ac.uk/29189/ (cit. on p. 43).

[23] J. Bankier and K. Gleason. «Institutional Repository Software Comparison». In: *UNESCO* 33.0 (2014) (cit. on pp. 25, 33).

[24] P. Barbosa. «UPBox: Armazenamento na Nuvem para Dados de Investigação da U . Porto». Master's Thesis. Faculdade de Engenharia da Universidade do Porto, 2013. URL: https://repositorio-aberto.up.pt/handle/10216/66874 (cit. on p. 7).

[25] G. Bartha and S. Kocsis. «Standardization of Geographic Data: The European INSPIRE directive». In: *European Journal of Geography* (2011), pp. 79–89 (cit. on pp. 7, 115, 148).

[26] N. Beagrie and B. Lavoie. *Keeping research data safe - KRSD2 Data Survey - Selection Criteria*. Tech. rep. Charles Beagrie Limited / JISC, 2009, pp. 1–8. URL: http://www.dlorg.eu/uploads/keepingresearchdatasafe2.pdf (cit. on p. 30).

[27] H. Beedham, J. Missen, and M. Palmer. *Assessment of UKDA and TNA Compliance with OAIS and METS Standards*. Tech. rep. UK Data Archive / JISC, 2004. URL: http://ie-repository.jisc.ac.uk/51/1/Report{_}oaismets.pdf (cit. on p. 30).

[28] A. Behr, T. T. Primo, and R. Vicari. «OBAA-LEME: A Learning Object Metadata Content Editor supported by Application Profiles and Educational Metadata Ontologies». In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* 3.1 (2014). ISSN: 2316-8889 (cit. on p. 20).

[29] F. Berman. «Got data?: a guide to data preservation in the information age». In: *Communications of the ACM* 51.12 (2008), pp. 50–56. ISSN: 0001-0782. DOI: 10.1145/1409360.1409376 (cit. on p. 1).

[30] G. Bernardo Cuenca, H. Ian, K. Yevgeny, and S. Ulrike. «Modular reuse of ontologies: theory and practice». In: *Journal of Artificial Intelligence Research* 31.1 (2008), pp. 273–318 (cit. on p. 20).

[31] T. Berners-Lee. *Linked Data—Design Issues*. 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html (cit. on pp. 31, 96, 110).

[32] T. Berners-Lee, J. Hendler, and O. Lassila. «The Semantic Web». In: *Scientific American*. Lecture Notes in Computer Science 284.5 (2001). Ed. by K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, pp. 34–43. ISSN: 00368733. DOI: 10.1038/scientificamerican0501-34 (cit. on p. 109).

[33] A. Bhardwaj, S. Bhattacherjee, and A. Chavan. «DataHub: Collaborative Data Science & Dataset Version Management at Scale». In: *7th Biennial Conference on Innovative Data Systems Research (CIDR'15) January*. Asilomar, California, USA, 2015 (cit. on p. 44).

[34] A. Bhardwaj, A. J. Elmore, U Chicago, D. Karger, S. Madden, E. Wu, and R. Zhang. «Collaborative Data Analytics with DataHub». In: *Proceedings of the VLDB Endowment* 8.12 (2015), pp. 1–4 (cit. on p. 44).

[35] E. P. Bontas, M. Mochol, and R. Tolksdorf. «Case studies on ontology reuse». In: *Proceedings of the 5th International Conference on Knowledge Management (IKNOW05)*. 2005 (cit. on pp. xviii, 18).

[36] C. Borgman. «The conundrum of sharing research data». In: *Journal of the American Society for Information Science and Technology* 63.6 (2012) (cit. on pp. 1, 3, 12, 31, 37, 77, 78, 109, 153).

[37] C. L. Borgman, J. C. Wallis, and N. Enyedy. «Little science confronts the data deluge: habitat ecology, embedded sensor networks, and digital libraries». In: *International Journal on Digital Libraries* 7.1-2 (2007), pp. 17–30 (cit. on p. 2).

[38] A. Boyko, J. Kunze, California Digital Library, J. Littman, L. Madden, Library of Congress, and B. Vargas. *The BagIt File Packaging Format (Vo.97)*. 2012. URL: https://tools.ietf.org/html/draft-kunze-bagit-06 (cit. on pp. 43, 169).

[39] J. S. Breese, D. Heckerman, and C. Kadie. «Empirical analysis of predictive algorithms for collaborative filtering». In: *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence* 461.8 (1998), pp. 43–52. ISSN: 15532712. DOI: 10.1111/j.1553-2712.2011.01172.x (cit. on p. 67).

[40] D. Brickley and R. V. Guha. *RDF Schema 1.1 W3C Recommendation*. 2014. DOI: 10.1016/B978-0-12-373556-0.00006-X. URL: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/ (cit. on p. 115).

[41] S. Brown, B. Rachel, and D. Kernohan. «Directions for Research Data Management in UK Universities». In: March (2015) (cit. on p. 1).

[42] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. «Learning to Rank Using Gradient Descent». In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML '05. New York, NY, USA: ACM, 2005, pp. 89–96. ISBN: 1-59593-180-5. DOI: 10.1145/1102351.1102363 (cit. on p. 60).

[43] C. S. Burns, A. Lana, and J. Budd. «Institutional repositories: exploration of costs and value». In: *D-Lib Magazine* 19.1/2 (2013). DOI: 10.1045/january2013 (cit. on p. 33).

[44] S. H. M. Butchart and M. et. al. Walpole. «Global Biodiversity: Indicators of Recent Declines». In: *Science* 328.5982 (2010), pp. 1164–1168. DOI: 10.1126/science.1187512 (cit. on p. 114).

[45] L. Candela, D. Castelli, P. Manghi, and A. Tani. «Data journals: A survey». In: *Journal of the Association for Information Science and Technology* 66.9 (2015), pp. 1747–1762. DOI: 10.1002/asi.23358 (cit. on pp. 3, 33).

[46] S. Carrier and L. Science. «The DRIADE project: Phased application profile development in support of open science». In: *Molecular Biology and Evolution* (2007), pp. 35–42 (cit. on p. 50).

[47] J. Castro, C. Ribeiro, and J. Rocha. «Designing an Application Profile Using Qualified Dublin Core: A Case Study with Fracture Mechanics Datasets». In: *Proceedings of the International Conference on Dublin Core and Metadata Applications 2013*. 2013, pp. 47–52 (cit. on pp. 18, 148).

[48] J. Castro, J. Rocha, and C. Ribeiro. «Creating lightweight ontologies for dataset description: Practical applications in a cross-domain research data management workflow». In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL and TPDL) DL 2014*. ACM Press, 2014 (cit. on pp. 7, 110, 145, 147, 163).

[49] J. Castro. «Estudo de utilização do Repositório de dados da Universidade do Porto». Masters' Thesis. Faculdade de Engenharia da Universidade do Porto, 2013. URL: https://repositorio-aberto.up.pt/handle/10216/68647 (cit. on pp. 5, 6, 8, 111, 163).

[50] J. Castro, D. Perrotta, R. Amorim, J. Rocha, and C. Ribeiro. «Ontologies for research data description: a design process applied to vehicle simulation». In: *Proceedings of the 9th Metadata and Semantics Research Conference (MTSR 2015)*. 2015 (cit. on pp. 110, 145, 147, 148, 163).

[51] D. C. Centre. *What is representation information?* http://www.dcc.ac.uk/resources/curation-lifecycle-model/lifecycle-model-faqs#node-1082-question-5. 2011 (cit. on p. 11).

[52] L. M. Chan and M. L. Zeng. «Metadata interoperability and standardization - A study of methodology part I: Achieving interoperability at the schema level». In: *D-Lib Magazine* 12.6 (2006), pp. 23–41. ISSN: 10829873. DOI: 10.1111/j.1468-0084.2006.00151.x (cit. on pp. xvii, 17, 18, 109).

[53] I. D. Citation, M. Working, G. Joan, C. D. Library, J. Ashton, B. Library, J. Brase, and D. P. Bracke. *DataCite Metadata Scheme for the Publication and Citation of Research Data*. Tech. rep. 2011 (cit. on pp. xix, 32).

[54] S. J. Coles, J. G. Frey, and C. L. Bird. «First steps towards semantic descriptions of electronic laboratory notebook records.» In: *Journal of Cheminformatics 2013* (2013), pp. 1–10 (cit. on p. 33).

[55] O Corcho. «Ontology based document annotation: trends and open research problems». In: *International Journal of Metadata, Semantics and Ontologies* 1.1 (2006), pp. 47–57 (cit. on pp. 111, 116).

[56] C. Coronel, S. Morris, and P. Rob. *Database systems: design, implementation, and management*. Ed. by Cengage Learning. 10th. 2012. ISBN: 1111969604 (cit. on p. xviii).

[57] E. Corrado. «The importance of open access, open source, and open standards for libraries». In: *Issues in science and technology librarianship* 42.Spring 2005 (2005), pp. 1–7 (cit. on pp. 80, 86).

[58] L Corti, V den Eynden, L Bishop, and M Woollard. *Managing and Sharing Research Data: A Guide to Good Practice*. SAGE Publications, 2014. ISBN: 9781446267264 (cit. on p. 38).

[59] Council of the Consultative Committee for Space Data Systems. *Reference Model for an Open Archival Information System (OAIS)*. Tech. rep. January. Consultative Committee for Space Data Systems, 2002. URL: http://nssdc.gsfc.nasa.gov/nost/isoas/us19/650x0{_}20010226rl.pdf (cit. on pp. xix, 28, 29, 34).

[60] K. Coyle and T. Baker. *Guidelines for Dublin Core Application Profiles*. 2009. URL: http://dublincore.org/documents/profile-guidelines/ (visited on 11/01/2015) (cit. on pp. 7, 85).

[61] R. Crow. *A guide to institutional repository software, 3rd Edition*. Tech. rep. August. Chain Bridge Group, 2004. URL: http://www.budapestopenaccessinitiative.org/pdf/OSI{_}Guide{_}to{_}IR{_}Software{_}v3.pdf (cit. on p. 32).

[62] M. D'Aquin, J. Euzenat, C. Le Duc, and H. Lewen. «Sharing and reusing aligned ontologies with Cupboard». In: *Proceedings of the fifth international conference on Knowledge capture (K-CAP '09)* (2009), p. 179. DOI: 10.1145/1597735.1597771 (cit. on p. 22).

[63] M. D'Aquin and H. Lewen. «Cupboard - A place to expose your ontologies to applications and the community». In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5554 LNCS (2009), pp. 913–918. ISSN: 03029743. DOI: 10.1007/978-3-642-02121-3{_}81 (cit. on p. 22).

[64] M. D'Aquin and E. Motta. «Watson, More Than a Semantic Web Search Engine». In: *Semantic Web* 2.0 (2011), pp. 55–63. ISSN: 15700844. DOI: 10.3233/SW-2011-0031 (cit. on p. 22).

[65] M. Day. «Integrating metadata schema registries with digital preservation systems to support interoperability: a proposal». In: *International Conference on Dublin Core and Metadata Applications (DC-2003), 2003-09-27 - 2003-10-02, Seattle, WA.* (2003) (cit. on p. 109).

[66] C. Desrosiers and G. Karypis. «A Comprehensive Survey of Neighborhood-based Recommendation Methods». In: *Recommender Systems Handbook* 54 (2011), pp. 107–144. ISSN: 14779234. DOI: 10.1007/978-0-387-85820-3{_}4 (cit. on p. 67).

[67] R. Devarakonda and G. Palanisamy. «Data sharing and retrieval using OAI-PMH». In: *Earth Science Informatics* 4.1 (2011), pp. 1–5. ISSN: 1865-0473. DOI: 10.1007/s12145-010-0073-0 (cit. on pp. 30, 31, 41).

[68] R. Devarakonda, G. Palanisamy, B. E. Wilson, and J. M. Green. «Mercury: reusable metadata management, data discovery and access system». In: *Earth Science Informatics* 3.1-2 (2010), pp. 87–94. ISSN: 1865-0473. DOI: 10.1007/s12145-010-0050-7 (cit. on p. 32).

[69] E. Duval, W. Hodgins, and S. Sutton. «Metadata principles and practicalities». In: *D-lib Magazine* 8.4 (2002), pp. 1–10. ISSN: 1082-9873. DOI: 10.1045/april2002-weibel (cit. on pp. 18, 19, 110).

[70] M. D. Ekstrand, R. T. John, and J. A. Konstan. «Collaborative filtering recommender systems». In: *Foundations and Trends in Human-Computer Interaction* 4.April (2011), p. 172. ISSN: 1551-3955. DOI: 10.1561/1100000009 (cit. on p. 67).

[71] Engineering and Physical Sciences Research Council. *EPSRC policy framework on research data - Scope and benefits*. 2015. URL: https://www.epsrc.ac.uk/about/standards/researchdata/scope/ (cit. on p. 1).

[72] European Commission. *Guidelines on Open Access to Scientific Publications and Research Data in Horizon 2020*. Tech. rep. December. 2013, pp. 1–14 (cit. on p. 1).

[73] European Commission. «Guidelines on Open Access to Scientific Publications and Research Data in Horizon 2020». In: December (2013), pp. 1–14 (cit. on pp. 4, 43).

[74] European Commission. *State of progress in the development of guidelines to express elements of the INSPIRE metadata implementing rules using ISO15836*. Tech. rep. European Commission, 2008, pp. 3–4. URL: http://inspire.ec.europa.eu/reports/ImplementingRules/metadata/MD{_}IR{_}and{_}DC{_}stateofprogress.pdf (cit. on p. 116).

[75] V. V. D. Eynden, L. Corti, L. Bishop, and L. Horton. *Managing and Sharing Data*. Third Edit. May. UK Data Archive University of Essex Wivenhoe Park Colchester Essex CO4 3SQ, 2011. ISBN: 1904059783 (cit. on pp. 37, 78).

[76] E Fay. «Repository software comparison: building digital library infrastructure at LSE». In: *Ariadne* 2009 (2010), pp. 1–11 (cit. on p. 25).

[77] A. Felfernig, L. Chen, and M. Mandl. «RecSys' 11 workshop on human decision making in recommender systems». In: *... conference on Recommender systems* (2011) (cit. on p. 126).

[78] B. S. Francesco Ricci, Lior Rokach. «Recommender Systems Handbook». In: *Recommender Systems Handbook*. 2011. Chap. 1, pp. 1–35. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3 (cit. on p. 58).

[79] R. Gattelli. «Gestão de dados de investigação no domínio da oceanografia biológica: criação e avaliação de um perfil de aplicação baseado em ontologia». Master's Thesis. Faculdade de Engenharia da Universidade do Porto, 2015. URL: https://repositorio-aberto.up.pt/handle/10216/79336 (cit. on p. 148).

[80] A. Goben and D. Salo. «Federal research: Requirements set to change». In: *College & Research Libraries News* 74 (2013), pp. 421–425. ISSN: 2150-6698 (cit. on p. 4).

[81] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. «Using collaborative filtering to weave an information tapestry». In: *Communications of the ACM* 35.12 (1992), pp. 61–70. ISSN: 00010782. DOI: 10.1145/138859.138867. arXiv: 39 (cit. on p. 58).

[82] S. Gordea, A. Lindley, and R. Graf. «Computing recommendations for long term data accessibility basing on open knowledge and linked data». In: *CEUR Workshop Proceedings*. Vol. 811. October. 2011, pp. 51–58 (cit. on p. 68).

[83] M. Gouveia. «DataNotes — Um sistema colaborativo para anotação de estruturas de diretórios». Master's Thesis. Faculdade de Engenharia da Universidade do Porto, 2013, p. 91. URL: https://repositorio-aberto.up.pt/handle/10216/66850 (cit. on p. 7).

[84] A. Goy, D. Magro, G. Petrone, C. Picardi, and M. Segnan. «Ontology-driven collaborative annotation in shared workspaces». In: *Future Generation Computer Systems* (2015). ISSN: 0167739X. DOI: 10.1016/j.future.2015.04.013 (cit. on pp. 72, 149).

[85] A. Green, S. Macdonald, and Robin Rice. *Policy-making for Research Data in Repositories: A Guide*. Tech. rep. May. DISC-UK DataShare Project, JISC, 2009. URL: http://www.edna.edu.au/edna/referral/browse/http://www.disc-uk.org/docs/guide.pdf (cit. on pp. 37, 48).

[86] J. Greenberg, S. Swauger, and E. M. Feinstein. «Metadata Capital in a Data Repository». In: *Proceedings of the International Conference on Dublin Core and Metadata Applications 2013*. Lisbon: Data Dryad Repository, 2013, pp. 140–150. DOI: doi:10.5061/dryad.8c1p6 (cit. on p. 150).

[87] T. Gruber. *Ontology*. 2009. URL: http://tomgruber.org/writing/ontology-definition-2007.htm (cit. on p. xviii).

[88] K. Haase. «Context for semantic metadata». In: *Proceedings of the 12th annual ACM international conference on Multimedia* (2004), pp. 204–211 (cit. on p. 14).

[89] T. Haerder and A. Reuter. «Principles of Transaction-oriented Database Recovery». In: *ACM Computing Surveys* 15.4 (1983), pp. 287–317. ISSN: 0360-0300. DOI: 10.1145/289.291 (cit. on p. xvii).

[90] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123814790, 9780123814791 (cit. on pp. 61, 136).

[91] H. Hanahoe, R. Baxter, A. Carter, J. Reetz, M. Riedel, R. Ritz, M. van de Sanden, and P. Wittenberg. «Second EUDAT Conference Report». 2014. URL: https://b2share.eudat.eu/record/201 (cit. on p. 96).

[92] B. Haslhofer and B. Schandl. «Interweaving OAI-PMH data sources with the linked data cloud». In: *International Journal of Metadata, Semantics and Ontologies* 1.5 (2010) (cit. on p. 32).

[93] B. Haslhofer and B. Schandl. «The OAI2LOD Server: Exposing OAI-PMH metadata as linked data». In: *International Workshop on Linked Data on the Web (LDOW2008), co-located with WWW 2008* (2008) (cit. on pp. 30–32).

[94] R. Heery and M. Patel. «Application profiles: mixing and matching metadata schemas». In: *Ariadne* 25 (2000) (cit. on pp. 18, 20, 93, 96, 109, 133).

[95] P. B. Heidorn. «Shedding Light on the Dark Data in the Long Tail of Science». In: *Library Trends* 57.2 (2008), pp. 280–299. ISSN: 1559-0682. DOI: 10.1353/lib.0.0036 (cit. on pp. 1–3, 13, 34).

[96] B. Heitmann and C. Hayes. «Using Linked Data to Build Open, Collaborative Recommender Systems». In: *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence* (2010), pp. 76–81. DOI: 10.1.1.174.2755 (cit. on p. 68).

[97] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. «Evaluating collaborative filtering recommender systems». In: *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), pp. 5–53. ISSN: 1046-8188. DOI: 10.1145/963770.963772. arXiv: 50 (cit. on p. 72).

[98] D. Herzig and B. Ell. «Semantic MediaWiki in Operation: Experiences with Building a Semantic Portal». In: *Lecture Notes in Computer Science* 6497 (2010), pp. 114–128 (cit. on p. 92).

[99] A. J. G. Hey and A. E. Trefethen. «The Data Deluge: An e-Science Perspective». In: *Grid Computing—Making the Global Infrastructure a Reality* (2003). Ed. by F. Berman, G. C. Fox, and A. J. G. Hey, pp. 809–824 (cit. on p. 1).

[100] A. Hey, S. Tansley, and K. Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009 (cit. on p. 1).

[101] S. Hodson. *ADMIRAL: A Data Management Infrastructure for Research Activities in the Life sciences*. Tech. rep. University of Oxford, 2011 (cit. on pp. 18, 43, 78, 79, 89, 96).

[102] T. Hofmann. «Collaborative filtering via gaussian probabilistic latent semantic analysis». In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03* v (2003), p. 259. DOI: 10.1145/860480.860483 (cit. on p. 62).

[103] J. Honrado, J. Alonso, P. Castro, and E. al. *The BIO_SOS metadata geoportal and the external quality of pre-existing datasets. Deliverable 4.5 of the FP7 project BIO_SOS. Appendix 1*. Tech. rep. 2010 (cit. on pp. 115, 121).

[104] I. Horrocks. «Ontologies and the semantic web». In: *Communications of the ACM* (2008) (cit. on p. xviii).

[105] N. Houssos, B. Joerg, and J. Dvořák. *OpenAIRE Guidelines for CRIS Managers*. Tech. rep. June. OpenAIRE, 2015. URL: http://doi.org/10.5281/zenodo.17065 (cit. on p. 51).

[106] N. Houssos, B. Jörg, J. Dvořák, P. Príncipe, E. Rodrigues, P. Manghi, and M. K. Elbæk. «OpenAIRE Guidelines for CRIS Managers: Supporting Interoperability of Open Research Information through Established Standards». In: *Procedia Computer Science* 33 (2014), pp. 33–38. ISSN: 18770509. DOI: 10.1016/j.procs.2014.06.006 (cit. on p. 51).

[107] J. Hoxha and A. Brahaj. «Open government data on the web: A semantic approach». In: *International Conference on Emerging Intelligent Data and Web Technologies (EIDWT)*. IEEE. Tirana: IEEE, 2011, pp. 107–113. ISBN: 978-1-4577-0840-4. DOI: 10.1109/EIDWT.2011.24 (cit. on p. 36).

[108] R. Hu and P. Pu. «Acceptance Issues of Personality-based Recommender Systems». In: *Proceedings of the Third ACM Conference on Recommender Systems (Recsys '09)*. New York, New York, USA: ACM, 2009, pp. 221–224. ISBN: 9781605584355. DOI: 10.1145/1639714.1639753 (cit. on p. 126).

[109] Y. Hu, C. Volinsky, and Y. Koren. «Collaborative filtering for implicit feedback datasets». In: *ICDM '08 Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. July. IEEE Computer Society, 2008, pp. 263–272. ISBN: 9780769535029. DOI: 10.1109/ICDM.2008.22 (cit. on p. 64).

[110] J. Hunter and C. Lagoze. «Combining RDF and XML schemas to enhance interoperability between metadata application profiles». In: *Proceedings of the tenth international conference on World Wide Web WWW 01* (2001), pp. 457–466. DOI: 10.1145/371920.372100 (cit. on p. 20).

[111] D. C. M. Initiative. *Dublin Core Metadata Element Set, Version 1.0: Reference Description*. 2012. URL: http://dublincore.org/documents/1998/09/dces/ (cit. on p. 12).

[112] International Organization for Standardization. *ISO 19115:2003 Geography Information - Metadata*. Tech. rep. 2003. URL: http://www.iso.org/iso/catalogue{_}detail.htm?csnumber=26020 (cit. on p. 119).

[113] E. K. Jacob. «Ontologies and the Semantic Web». In: *Bulletin of the American Society for Information Science and Technology* 29.4 (2003), pp. 19–22. ISSN: 1550-8366. DOI: 10.1002/bult.283 (cit. on p. xviii).

[114] L. Jahnke, A. Asher, and S. D. C. Keralis. «The Problem of Data». In: *Council on Library and Information Resources* (2012). Ed. by Council on Library and Information Resources, pp. 1–43 (cit. on pp. 3, 78).

[115] R. Jin and L. Si. «A study of methods for normalizing user ratings in collaborative filtering». In: *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04* (2004), p. 568. DOI: 10.1145/1008992.1009124 (cit. on pp. 60–62).

[116] R. Jin, L. Si, C. Zhai, and J. Callan. «Collaborative filtering with decoupled models for preferences and ratings». In: *Proc. of the twelfth international conference on Information and knowledge management - CIKM '03* (2003), p. 309. DOI: 10.1145/956919.956922 (cit. on p. 62).

[117] T. Joachims, L. Granka, and B. Pan. «Accurately interpreting clickthrough data as implicit feedback». In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2005), pp. 154–161. DOI: 10.1145/1076034.1076063 (cit. on pp. 64, 72, 135, 149).

[118] P Johnston. «Metadata and interoperability in a complex world». In: *Ariadne* 37 (2003), pp. 1–5 (cit. on p. 18).

[119] R. Jones and M. Ramsay. *A Technical Review of Open Access Repository Registries*. Tech. rep. July. Cottage Labs for UKOLN, 2011. URL: http://ie-repository.jisc.ac.uk/612/ (cit. on p. 51).

[120] S. Jones. «Data Audit Framework Development (DAFD) Project Final Report». In: (2009) (cit. on p. 30).

[121] S. Jones, S. Ross, and R. Ruusalepp. *Data Audit Framework Methodology*. Tech. rep. HATII, University of Glasgow, 2009, p. 70. URL: http://www.data-audit.eu/DAF{_}Methodology.pdf (cit. on p. 30).

[122] B. Jörg. «CERIF: The common European research information format model». In: *Data Science Journal* 9.July (2010), pp. 24–31 (cit. on p. 110).

[123] H. Khrouf and R. Troncy. «Hybrid event recommendation using linked data and user diversity». In: *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13* (2013), pp. 185–192. DOI: 10.1145/2507157.2507171 (cit. on p. 68).

[124] D. A. Koutsomitropoulos, G. E. Paloukis, and T. S. Papatheodorou. «Semantic Application Profiles: A Means to Enhance Knowledge Discovery in Domain Metadata Models». In: *Metadata and Semantics* (2009), pp. 23–33. DOI: 10.1007/978-0-387-77745-0{_}3 (cit. on p. 20).

[125] D. Koutsomitropoulos, G. D. Solomou, A. D. Alexopoulos, and T. S. Papatheodorou. «Digital Repositories and the Semantic Web: Semantic Search and Navigation for DSpace (presentation)». In: *DSUG 09 Gothenburg* (2009) (cit. on p. 20).

[126] E. M. Krause, E. Clary, J. Greenberg, and A. Ogletree. «Evolution of an Application Profile: Advancing Metadata Best Practice s through the Dryad Data Repository». In: *Procedings of the International Conference on Dublin Core and Metadata Applications 2015*. 2015, pp. 63–75 (cit. on p. 50).

[127] J. Kučera, D. Chlapek, and J. Mynarz. *Czech CKAN Repository as Case Study in Public Sector Data Cataloging*. Tech. rep. 2012, pp. 95–107. URL: http://www.cssi.cz/cssi/czech-ckan-repository-case-study-public-sector-data-cataloging (cit. on p. 36).

[128] C. Lagoze, H. V. D. Sompel, M. Nelson, and S. Warner. *The Open Archives Initiative Protocol for Metadata Harvesting*. 2002 (cit. on pp. 30, 37, 41).

[129] O. Lassila and D. McGuinness. «The role of frame-based representation on the semantic web». In: *Linköping Electronic Articles in Computer and Information Science* 6.005 (2001) (cit. on p. 111).

[130] D. Lathrop and L. Ruma. *Open Government: Collaboration, Transparency, and Participation in Practice*. 1st. O'Reilly Media, Inc., 2010. ISBN: 0596804350, 9780596804350 (cit. on p. xviii).

[131] S. Leonelli, D. Spichtinger, and B. Prainsack. «Sticks and carrots: encouraging open science at its source». In: *Geo: Geography and Environment* (2015). ISSN: 20544049. DOI: 10.1002/geo2.2 (cit. on p. 3).

[132] H. Li. «A short introduction to Learning to Rank». In: *IEICE Transactions on Information and Systems* E94-D.1 (2011), pp. 1–9. ISSN: 0916-8532. DOI: 10.1587/transinf.E94.D.1 (cit. on p. 167).

[133] Y.-f. Li, G. Kennedy, F. Davies, and J. Hunter. «PODD-An Ontology-centric Data Management System for Scientific Research». In: *Proceedings of the International Conference on Asian Digital Libraries 2010*. Gold Coast, Queensland, Australia, 2010. DOI: 10.1007/978-3-642-13654-2{_}22 (cit. on p. 86).

[134] Y.-f. Li, G. Kennedy, F. Ngoran, P. Wu, and J. Hunter. «An Ontology-centric Architecture for Extensible Scientific Data Management Systems». In: *Future Generation Computer Systems* 29.2 (2013), pp. 1–38 (cit. on p. 86).

[135] Library of Congress. *METS: An Overview & Tutorial*. 2003. URL: https://www.loc.gov/standards/mets/METSOverview.v2.html (visited on 01/01/2016) (cit. on p. 16).

[136] G. Linden, B. Smith, and J. York. «Amazon.com recommendations: Item-to-item collaborative filtering». In: *IEEE Internet Computing* 7.1 (2003), pp. 76–80. ISSN: 10897801. DOI: 10.1109/MIC.2003.1167344 (cit. on p. 57).

[137] A. Lomba, C. Guerra, J. Alonso, J. P. Honrado, R. Jongman, and D. McCracken. «Mapping and monitoring High Nature Value farmlands: Challenges in European landscapes.» In: *Journal of environmental management* 143C (2014), pp. 140–150. ISSN: 1095-8630. DOI: 10.1016/j.jenvman.2014.04.029 (cit. on p. 114).

[138] P. Lord and A. Macdonald. *Data curation for e-Science in the UK: an audit to establish requirements for future curation and provision*. Tech. rep. JISC, 2003, pp. 1–84. URL: http://www.ukoln.ac.uk/projects/ebank-uk/curation/ (cit. on pp. xviii, 3, 12).

[139] P. Lord, A. Macdonald, L. Lyon, and D. Giaretta. «From Data Deluge to Data Curation». In: *Proceedings of the 3th UK e-Science All Hands Meeting*. 2004, pp. 371–375 (cit. on p. 1).

[140] C. A. Lynch. «Institutional repositories: essential infrastructure for scholarship in the digital age». In: *Portal: Libraries and the Academy* 3.2 (2003), pp. 327–336 (cit. on p. 1).

[141] L. Lyon. *Dealing with Data: Roles, Rights, Responsibilities and Relationships*. 2007. URL: http://www.ukoln.ac.uk/ukoln/staff/e.j.lyon/reports/dealing{_}with{_}data{_}report-final.pdf (cit. on pp. 3, 12, 27, 33, 37, 41).

[142] J. Madin, S. Bowers, M. Schildhauer, S. Krivov, D. Pennington, and F. Villa. «An ontology for describing and synthesizing ecological observation data». In: *Ecological Informatics* 2.3 (2007), pp. 279–296. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2007.05.004 (cit. on pp. xxi, 110, 111).

[143] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521865719, 9780521865715 (cit. on pp. 5, 59, 60).

[144] L. Martinez-Uribe. *Using the Data Audit Framework: an Oxford case study*. Tech. rep. Oxford Digital Repositories Steering Group, JISC, 2009. URL: http://www.disc-uk.org/publications.html (cit. on p. 78).

[145] L. Martinez-Uribe and S. Macdonald. «User engagement in research data curation». In: *Proceedings of the 13th European conference on Research and advanced technology for digital libraries*. Vol. 5714. Springer, 2009. Chap. 30, pp. 309–314 (cit. on pp. 25, 77, 78).

[146] A. Mauthe and P. Thomas. *Professional Content Management Systems: Handling Digital Media Assets*. Wiley, 2005. ISBN: 9780470855430 (cit. on p. xvii).

[147] D. L. McGuinness and F. van Harmelen. *OWL web ontology language overview - W3C Recommendation*. 2004. URL: https://www.w3.org/TR/owl-features/ (visited on 01/01/2016) (cit. on p. 80).

[148] M. McNutt. «Improving Scientific Communication». In: *Science* 342.6154 (2013), p. 13. DOI: 10.1126/science.1246449 (cit. on p. 1).

[149] P. Melville, R. J. Mooney, and R. Nagarajan. «Content-boosted collaborative filtering for improved recommendations». In: *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*. 2002, pp. 187–192. ISBN: 0262511290. DOI: 10.1.1.16.4936. arXiv: 86 (cit. on pp. 66, 67, 134).

[150] J. C. Molloy. «The Open Knowledge Foundation: Open Data Means Better Science». In: *PLoS Biol* 9.12 (2011), e1001195. DOI: 10.1371/journal.pbio.1001195 (cit. on p. xviii).

[151] A. Morgana and A. A. Baptista. «The use of Application Profiles and metadata schemas by digital repositories: findings from a survey». In: *Proceedings of the International Conference on Dublin Core and Metadata Applications 2015* (2015), pp. 146–157 (cit. on p. 150).

[152] M. Nagamori and S. Sugimoto. «A Metadata Schema Registry as a Tool to Enhance Metadata Interoperability». In: *Mitsuharu Nagamori and Shigeo Sugimoto* 3.1 (2006) (cit. on p. 14).

[153] National Information Standards Organization. *Understanding Metadata*. Ed. by NISO Press. Bethesda, 2004. ISBN: 1880124629. DOI: 10.1017/S0003055403000534. arXiv: 4 (cit. on pp. 13, 16).

[154] National Science Foundation. «Grants.Gov Application Guide A Guide for Preparation and Submission of NSF Applications via Grants.gov». 2011 (cit. on pp. 1, 4, 37).

[155] B. Nelson. «Data Sharing: Empty archives». In: *Nature* 461.September (2009) (cit. on pp. 4, 77).

[156] M. Nilsson and J. Brase. «The LOM RDF binding - principles and implementation». In: *Proceedings of the Third Annual ARIADNE conference, Leuven Belgium, 2003*. 2003, pp. 1–7 (cit. on p. 17).

[157] OECD. *Controlled Vocabulary*. 2005. URL: http://stats.oecd.org/glossary/detail.asp?ID=6260 (cit. on pp. xvii, 11).

[158] D. Oard and J. Kim. «Implicit feedback for recommender systems». In: *Procedings of the AAAI workshop on recommender systems* (1998) (cit. on pp. 64, 135).

[159] Open Knowledge Foundation. *CKAN documentation - Release 2.2a*. 2014. URL: http://ckan.readthedocs.org/en/new-rtd-default-docs-theme/ (visited on 01/01/2016) (cit. on p. 78).

[160] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. «Top-N recommendations from implicit feedback leveraging linked open data». In: *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13* (2013), pp. 85–92. ISSN: 16130073. DOI: 10.1145/2507157.2507172 (cit. on p. 69).

[161] L. Page, S. Brin, R. Motwani, and T. Winograd. «The PageRank Citation Ranking: Bringing Order to the Web». In: *World Wide Web Internet And Web Information Systems* 54.1999-66 (1998), pp. 1–17. ISSN: 1752-0509. DOI: 10.1.1.31.1768. arXiv: 1111.4503v1 (cit. on p. 60).

[162] N. Paskin. «Digital Object Identifier (DOI ®) System». In: *Encyclopedia of Library and Information Sciences, Third Edition* August 2013 (2011), pp. 37–41. ISSN: 978-0-8493-9712-7. DOI: 10.1081/E-ELIS3-120044418 (cit. on p. 32).

[163] A. Passant. «dbrec - Music Recommendations Using DBpedia». In: *9th International Semantic Web Conference (ISWC2010)*. Vol. 1380. 2007, pp. 1–16 (cit. on p. 68).

[164] M. Pazzani, J. Muramatsu, and D. Billsus. «Syskill & Webert: Identifying interesting web sites». In: *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1* (1996), pp. 54–61. DOI: citeulike-article-id:1188705 (cit. on p. 63).

[165] H. M. Pereira, J. Belnap, N. Brummitt, B. Collen, H. Ding, M. Gonzalez-Espinosa, J. Gregory, R.D. Honrado, R. H. Jongman, R. Julliard, L. McRae, V. Proença, P. Rodrigues, M. Opige, J. P. Rodriguez, D. S. Schmeller, C. van Swaay, and C. Vieira. «Global biodiversity monitoring». In: *Frontiers in Ecology and the Environment* 8 (2010), pp. 459–460 (cit. on p. 114).

[166] B. J. Pine. *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, 1993. ISBN: 9780875843728 (cit. on p. 57).

[167] B. J. Pine and J. H. Gilmore. *The Experience Economy: Work is Theatre & Every Business a Stage*. Harvard Business School Press. Harvard Business School Press, 1999. ISBN: 9780875848198 (cit. on p. 57).

[168] M. Pinho. «Modelo de Replicação para a Preservação de Dados em Repositórios». Masters' Thesis. Faculdade de Engenharia da Universidade do Porto, 2012. URL: https://repositorio-aberto.up.pt/handle/10216/68432 (cit. on p. 7).

[169] A. Pipitone and R. Pirrone. «A framework for automatic semantic annotation of Wikipedia articles». In: *6th Workshop on Semantic Web Applications and Perspectives*. 2010 (cit. on p. 92).

[170] H. Piwowar, R. Day, and D. Fridsma. «Sharing detailed research data is associated with increased citation rate». In: *PLoS ONE* 2.3 (2007) (cit. on pp. 3, 34).

[171] H. Piwowar and T. Vision. «Data reuse and the open data citation advantage». In: *PeerJ* 1 (2013), e175. ISSN: 2167-8359. DOI: 10.7717/peerj.175 (cit. on pp. 3, 34).

[172] B. Plale. «Managing the long tail of science: data and communities». In: *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond*. 2012. ISBN: 9781450316026. DOI: 10.1145/2335755.2335866 (cit. on p. 2).

[173] I. Pôças, J. Gonçalves, B. Marcos, J. Alonso, P. Castro, and J. P. Honrado. «Evaluating the fitness for use of spatial data sets to promote quality in ecological assessment and monitoring». In: *International Journal of Geographical Information Science* June (2014), pp. 1–16. ISSN: 1365-8816. DOI: 10.1080/13658816.2014.924627 (cit. on pp. 114, 115, 121).

[174] M. Poschen, J. Finch, R. Procter, M. Goff, M. McDerby, S. Collins, J. Besson, L. Beard, and T. Grahame. «Development of a Pilot Data Management Infrastructure for Biomedical Researchers at University of Manchester – Approach, Findings, Challenges and Outlook of the MaDAM Project». In: *International Journal of Digital Curation* 7 (2012), pp. 110–122. ISSN: 1746-8256. DOI: 10.2218/ijdc.v7i2.234 (cit. on p. 43).

[175] L. Pouchard, L. Cinquini, and G. Strand. «The earth system Grid discovery and semantic web technologies». In: *Workshop for Semantic Web Technologies for Searching and Retrieving Scientific Data - 2nd International Semantic Web Conference*. 2003 (cit. on p. 110).

[176] K. L. Priddy and P. E. Keller. *Artificial Neural Networks: An Introduction*. Tutorial Text Series. Society of Photo Optical, 2005, p. 16. ISBN: 9780819459879 (cit. on p. 61).

[177] P. Príncipe and J. Schirrwagen. *OpenAIRE guidelines for data source managers: Aiming for metadata harmonization*. 2015. DOI: 10.5281/zenodo.6918. URL: http://hdl.handle.net/1822/35655 (cit. on p. 54).

[178] J. Qin and K. Li. «How Portable Are the Metadata Standards for Scientific Data? A Proposal for a Metadata Infrastructure». In: *Proceedings of the DC-2013 Conference*. 2013, pp. 25–34 (cit. on p. 111).

[179] J. Qin, A. Ball, and J. Greenberg. «Functional and Architectural Requirements for Metadata: Supporting Discovery and Management of Scientific Data». In: *Proceedings of the International Conference on Dublin Core and Metadata Applications 2012*. 2012, pp. 62–71 (cit. on p. 111).

[180] K. Rafes and C. Germain. «A platform for scientific data sharing». In: *BDA2015 Bases de Données Avancées*. 2015 (cit. on p. 44).

[181] A. U. Rahman, G. David, and C. Ribeiro. «Model Migration Approach for Database Preservation». In: *The Role of Digital Libraries in a Time of Global Change, 12th International Conference on Asia-Pacific Digital Libraries (ICADL) 2010*. Gold Coast, Australia, 2010, pp. 81–90. DOI: 10.1007/978-3-642-13654-2{_}10 (cit. on p. 80).

[182] E. Rajabi, Miguel-Angel Sicilia, Hannes Ebner, Matthias Palmer, and Salvador Sanchez. *Recommendation on exposing IEEE LOM as Linked Data 1.0 (second version)*. 2014. URL: http://data.opendiscoveryspace.eu/ODS{_}LOM2LD/ODS{_}SecondDraft.html (visited on 12/01/2015) (cit. on p. 17).

[183] J. Ramalho, M. Ferreira, L. Faria, and R. Castro. «RODA and CRiB a service-oriented digital repository». In: *Proceedings of the 5th International Conference on Preservation of Digital Objects (iPRES 2008)* (2008) (cit. on p. 34).

[184] P. Resnick and H. R. Varian. «Recommender Systems». In: *Communications of the ACM* 40.3 (1997), pp. 56–58. ISSN: 0001-0782. DOI: 10.1145/245108.245121. arXiv: 1202.1112v1 (cit. on pp. 57, 58, 133).

[185] D. Retelny, S. Robaszkiewicz, A. To, W. Lasecki, J. Patel, N. Rahmati, T. Doshi, M. Valentine, and M. Bernstein. «Expert crowdsourcing with flash teams». In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (2014). DOI: 10.1145/2642918.2647409 (cit. on p. 146).

[186] C. Ribeiro and M. E. M. Fernandes. «Data Curation at U. Porto: Identifying current practices across disciplinary domains». In: *IASSIST Quarterly* 35 (2011), pp. 14–17 (cit. on p. 147).

[187] F Ricci and B Shapira. *Recommender systems handbook*. Springer, 2011. ISBN: 978-0-387-85819-7 (cit. on pp. 63, 66).

[188] J. Rocha, J. Barbosa, M. Gouveia, C. Ribeiro, and J. Correia Lopes. «UPBox and DataNotes: a collaborative data management environment for the long tail of research data». In: *iPres 2013 Conference Proceedings*. 2013 (cit. on pp. 6, 27, 79, 93).

[189] J. Rocha, J. Castro, C. Ribeiro, and J. Correia Lopes. «Dendro: collaborative research data management built on linked open data». In: *Proceedings of the 11th European Semantic Web Conference* (2014) (cit. on pp. 6, 78, 95, 96, 133).

[190] J. Rocha, J. Castro, C. Ribeiro, and J. Correia Lopes. «The Dendro research data management platform: Applying ontologies to long-term preservation in a collaborative environment». In: *iPres 2014 Conference Proceedings*. 2014 (cit. on pp. 6, 78, 95, 96, 121, 133).

[191] J. Rocha, J. Castro, C. Ribeiro, J. Honrado, A. Lomba, and J. Gonçalves. «Beyond INSPIRE: An ontology for biodiversity metadata records». In: *Proceedings of the 10th International Workshop on Ontology Content (OntoContent 2014)*. Ed. by Springer. 2014 (cit. on pp. 7, 110, 148, 163).

[192] J. Rocha, C. Ribeiro, and J. Correia Lopes. «Managing multidisciplinary research data: Extending DSpace to enable long-term preservation of tabular datasets». In: *iPres 2012 Conference*. 2012, pp. 105–108 (cit. on pp. 38, 77, 86).

[193] J. Rocha, C. Ribeiro, and J. Correia Lopes. «Managing research data at U.Porto: requirements, technologies and services». In: *Innovations in XML Applications and Metadata Management: Advancing Technologies* IGI Global (2012) (cit. on pp. 3, 78, 86, 134).

[194] J. Rocha, C. Ribeiro, and J. Correia Lopes. «Semi-automated application profile generation for research data assets». In: *Metadata and Semantics Research Conference* (2012) (cit. on pp. 5, 18, 69, 134, 144).

[195] J. Rocha, C. Ribeiro, and J. Correia Lopes. «UPBox e DataNotes: um ambiente de suporte à gestão colaborativa de dados científicos». In: *InCID: Revista de Ciência da Informação e Documentação* 4.2 (2013). ISSN: 2178-2075 (cit. on pp. 6, 79).

[196] J. Rocha, C. Ribeiro, and J. Correia Lopes. «UPData—A Data Curation Experiment at U.Porto using DSpace». In: *iPRES 2011 Proceedings*. 2011, pp. 224–227 (cit. on pp. 5, 25, 38, 77, 134).

[197] J. Rocha, C. Ribeiro, and J. Correia Lopes. «Ontology-based multidomain metadata for research data management using triple stores». In: *Proceedings of the 18th International Database Engineering & Applications Symposium*. 2014. ISBN: 9781450326278 (cit. on pp. 6, 34, 39, 78, 79, 96).

[198] M. Salvadores, P. R. Alexander, M. A. Musen, and N. F. Noy. «BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF.» In: *Semantic web* 4.3 (2013), pp. 277–284. ISSN: 1570-0844. DOI: 10.3233/SW-2012-0086 (cit. on p. 22).

[199] J. B. Schafer, J. Konstan, and J. Riedl. «E-commerce recommendation applications». In: *Data Mining and Knowledge Discovery* 5.1-2 (2001), pp. 115–153. ISSN: 13845810. DOI: 10.1007/978-1-4615-1627-9{_}6. arXiv: 0010006 [cs.LG] (cit. on p. 66).

[200] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. «Methods and metrics for cold-start recommendations». In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02)* 46 (2002), pp. 253–260. ISSN: 01635840. DOI: 10.1145/564376.564421 (cit. on pp. 67, 134).

[201] Science Magazine. «Dealing with data. Challenges and opportunities. Introduction.» In: *Science (New York, N.Y.)* 331.6018 (2011), pp. 692–3. ISSN: 1095-9203. DOI: 10.1126/science.331.6018.692 (cit. on pp. 2, 3).

[202] G. Shani and A. Gunawardana. «Evaluating recommendation systems». In: *Recommender systems handbook*. Ed. by P. B. K. Francesco Ricci, Lior Rokach, Bracha Shapira. 1st. Springer US, 2011. Chap. 7, pp. 1–41. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3 (cit. on pp. 71, 127).

[203] G. Shani and Gunwa. *Evaluating Recommendation Systems*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Boston, MA: Springer US, 2011. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3 (cit. on pp. 72, 146).

[204] D. Shotton. «The JISC UMF DataFlow Project : Introduction to DataStage». In: *Technical Report* (2012) (cit. on pp. 18, 78, 79, 89).

[205] A. Singhal, C. Buckley, and M. Mitra. «Pivoted Document Length Normalization». In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '96. New York, NY, USA: ACM, 1996, pp. 21–29. ISBN: 0-89791-792-8. DOI: 10.1145/243199.243206 (cit. on pp. 61, 144).

[206] R. Sinha and K. Swearingen. «Comparing Recommendations Made by Online Systems and Friends». In: *DELOS Workshop on Personalisation and Recommender Systems in Digital Libraries* (2001) (cit. on p. 126).

[207] R. Sinha and K. Swearingen. «The role of transparency in recommender systems». In: *CHI '02 Extended abstracts on Human factors in computing systems* (2002), p. 830. DOI: 10.1145/506443.506619. arXiv: 126 (cit. on pp. 131, 151).

[208] I. Soboroff and C. Nicholas. «Combining content and collaboration in text filtering». In: *International Joint Conferences on Artificial Intelligence* 99 (1999), pp. 86–91. ISSN: 03640213. DOI: 10.3115/1118935.1118938 (cit. on p. 66).

[209] S. Soiland-Reyes, M. Gamble, and R. Haines. *Research Object Bundle 1.0*. 2014. URL: https://researchobject.github.io/specifications/bundle/{\#}json-structure (visited on 11/01/2015) (cit. on p. 169).

[210] L. N. Soldatova and R. D. King. «An ontology of scientific experiments.» In: *Journal of the Royal Society, Interface / the Royal Society* 3.11 (2006), pp. 795–803. ISSN: 1742-5689. DOI: 10.1098/rsif.2006.0134 (cit. on p. 110).

[211] C. Strasser. *Research data management: A primer publication of the National Information Standards Organization*. Tech. rep. Baltimore: National Information Standards Organization (NISO), 2015, pp. 0–16. URL: http://www.niso.org/publications/press/researchdata/ (cit. on p. 26).

[212] S. Strickroth and N. Pinkwart. ≪High quality recommendations for small communities: the case of a regional parent network≫. In: *Proceedings of the sixth ACM conference on Recommender systems* (2012), pp. 107–114 (cit. on pp. 64, 72, 126, 135, 149, 166).

[213] A. Swan and S. Brown. ≪The skills, role and career structure of data scientists and curators: An assessment of current practice and future needs. Report to the JISC≫. 2008. URL: http://eprints.soton.ac.uk/266675/ (cit. on p. 37).

[214] K. Swearingen and R. Sinha. ≪Beyond Algorithms Beyond Algorithms: An HCI Perspective on Recommender Systems≫. In: *ACM SIGIR 2001 Workshop on Recommender Systems (2001)* (2001), pp. 1–11. DOI: 10.1.1.23.9764 (cit. on pp. 63, 131, 135).

[215] G. Takács and D. Tikk. ≪Alternating least squares for personalized ranking≫. In: *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12* (2012), p. 83. DOI: 10.1145/2365952.2365972 (cit. on p. 64).

[216] The EUDAT Consortium. ≪Towards a Collaborative Data Infrastructure≫. In: *Technical Brochure* October (2011), p. 283304 (cit. on p. 44).

[217] The University of Edinburgh. *Edinburgh DataShare: Depositor's User Guide*. 2015. URL: http://www.ed.ac.uk/files/atoms/files//datashare-may2015.pdf (cit. on p. 48).

[218] P.-y. Vandenbussche, G. A. Atemezing, and B. Vatant. ≪Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web≫. In: *IOS press* 1 (2014), pp. 1–5 (cit. on p. 21).

[219] S. Vermaaten. ≪A Checklist and a Case for Documenting PREMIS-METS Decisions in a METS Profile≫. In: *D-Lib Magazine* 16.9/10 (2010), pp. 1–13. DOI: 10.1045/september2010-vermaaten (cit. on p. 16).

[220] M Völkel, M Krötzsch, and D Vrandecic. ≪Semantic wikipedia≫. In: *Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006)* (2006) (cit. on p. 90).

[221] K. Weller and K. E. Kinder-kurlanda. ≪Uncovering the Challenges in Collection, Sharing and Documentation: The Hidden Data of Social Media Research?≫ In: *Proceedings of the Ninth International AAAI Conference on Web and Social Media* 2014 (2015), pp. 28–37 (cit. on p. 3).

[222] C. Willis, J. Greenberg, and H. White. ≪Analysis and Synthesis of Metadata Goals for Scientific Data≫. In: *Journal of the Association for Information Science and Technology* 63.8 (2012), pp. 1505–1520. DOI: 10.1002/asi (cit. on p. 39).

[223] J. Winn. ≪Open data and the academy: An evaluation of CKAN for research data management≫. In: *International Association dor Social Science Information Services and Technology* (2013) (cit. on p. 36).

[224] M. S. Woodley. ≪Crosswalks , Metadata Harvesting, Federated Searching, Metasearching: Using Metadata to Connect Users and Information≫. In: *Introduction to Metadata* (2008), pp. 1–25 (cit. on p. 17).